



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

이학박사 학위논문

An Improved Authentication and Key Exchange Scheme

(개선된 인증과 키분배 기법)

2014년 2월

서울대학교 대학원

수리과학부

김경국

An Improved Authentication and Key Exchange Scheme

(개선된 인증과 키분배 기법)

지도 교수 김명환

이 논문을 이학박사 학위논문으로 제출함
2014 년 2 월

서울대학교 대학원
수리과학부
김경국

김경국의 박사 학위논문을 인준함
2014년 2월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

Abstract

An Improved Authentication and Key Exchange Scheme

Kyung-Kug Kim

Department of Mathematical Sciences

The Graduate School

Seoul National University

Nowadays, anonymity property of user authentication scheme becomes important. From 2003, Park *et al.*, Juang *et al.*, and other researchers proposed a useful, secure and efficient authenticated-key exchange scheme. However, There schemes did not provide the useful methods against the various efficient attacks. They argued that they provided the identity privacy-mutual authentication-half-forward secrecy. But their suggestions have limited solutions. So we have researched the about 30 papers and suggested an improved authentication and key exchange scheme. Then, we show that the proposed scheme is secure against the various attacks methods (linear attack, inverse, dictionary, MTMD attacks etc).

Keywords: Cryptographic, Traitor tracing, revoke scheme, linear attack

Student Number: 2004-30103

Contents

| | |
|---|----|
| Chapter 1 Introduction | 6 |
| 1.1 Motivation | 6 |
| 1.2 Organization | 8 |
| Chapter 2 Secure Authenticated Key Exchange | 11 |
| 2.1 AKE Security | 11 |
| 2.2 Protocol Attack Types | 17 |
| Chapter 3 Secure Authenticated Key Exchange | 19 |
| 3.1 The Authentication Key Protocol | 19 |
| 3.2 General Security-Analysis Discussion | 26 |
| Chapter 4 Authenticated Key Exchange Protocol | 40 |
| 4.1 The Improved AKE | 41 |
| 4.2 An Improved Anonymous AKE Scheme | 62 |
| Chapter 5 Conclusion | 75 |

| | |
|--------------------|----|
| Bibliography | 77 |
|--------------------|----|

| | |
|----------------|----|
| Abstract | 87 |
|----------------|----|

List of Tables

| | | |
|-----------|--|----|
| Table 4.1 | Computational cost(related protocols)..... | 60 |
| Table 4.2 | Comparision of functionality between related protocols | 60 |
| Table 4.3 | Security Analysis..... | 73 |
| Table 4.4 | Efficiency alalysis | 67 |
| Table 4.5 | Security alalysis | 69 |

List of Figures

| | | |
|-----------|--|----|
| Figure3.1 | AKE Scheme of Huang-Wei's protocol | 22 |
| Figure3.2 | Attack Scheme for the AKE..... | 24 |
| Figure3.3 | Shi et al. 's protocol..... | 31 |
| Figure4.1 | LCW protocol | 43 |
| Figure4.2 | A MITM-attack on LCW protocol | 47 |
| Figure4.3 | Our Improved AKE protocol | 53 |
| Figure4.4 | Proposed protocol..... | 64 |

1. Introduction

1.1 Motivation

For establishing secure communications over unsafe public networks, authenticating of communicating entities and keeping transmitted datum unexposed are essential. When two or more parties participate to finish works in any desired secure way, even in the situation that there are adversaries, secure communication issues in unsafe open networks pose challenges [2].

These open networks are exposed to many spiteful attacks that some times result in leakage, change and destruction of significant information caused by the tampers of devices and the insecurity of remote communication. Simultaneously the problem catches the security demands of many computing applications that deal with sensitive information through antagonistic distributed environments and its significance increase given by the present general trend in current system design towards regionalized construction with least trust presumptions. To patch existing protocols and designing future protocols, it is important to understand how Authenticated key exchange (AKE) protocols fail [1, 2, 3, 4, 5, 6].

AKE is used as an opening procedure to give authority to communication members through secret key establishment to offer security services in open communication networks. In an authentication procedure, any user is asked to proffer secret datum same as challenge response values for validation. With session key, server provides security services and performs validation. With AKE procedure, the network-resources can be preserved by authenticated rightful communication members. The data integrity and information

confidentiality can also be credited by using the prearranged secret keys for message authentication and encryption. Therefore, the AKE is relevant to the network security: network resource, data integrity, information confidentiality and all other attacks.

In many fields of computing for solving problems of secure communication AKE protocols have been advanced, but to ensure security, it is a challenging task as ever.

New kinds of applications and security threats are able to be introduced as the Internet comes to be more universally accessible. Thus, password or secret key security will be endangered since new systems with Internet-support and computer-embedded have a regionalized or distributed architecture. The AKE protocol is geared to check communication members and produce secret keys with reciprocal trust. AKE procedure is inevitable to provide security because the reciprocal trust is to protect the communication members. To reduce security risk and increase security strength, we offer the building blocks required for AKE protocols. We make security decisions to depend on anecdotal heuristics, unproven evidences and expert opinions.

This dissertation offers an overall understand of AKE protocols. We research various malicious attacks on remote systems and analyze the security of earlier solutions for AKE meeting security threats on the AKE for session begin protocol, one-time-password AKE and hash-based protocols. We find out drawbacks and potential risks of the protocols. We suggest advancements on the earlier AKE protocols to set up secure session keys between communication parties against malevolent attacks for open unreliable networks using encryption procedure, one-time password and hash chain drawing lower operation and better security. Also, we bring advancements of earlier the authenticated key agreement protocols that establish a secure and

efficient for normal purpose AKE over unreliable adversarial network environments. Our advanced authenticated protocols are on the base of symmetric encryption and cryptographic hash function: (1) elaborate authenticated key agreement protocol protecting user privacy, (2) secure and efficient AKA protocol protecting user integrity and privacy.

From this research, we also suggest and analyze as follows: First, AKE security malevolent attacks into disparate parts for remote system, reconsider and analyze recent suggestions for AKE meeting security threats.

Second, En-route schemes for securing previous AKE using encryption (symmetric techniques), one-time password and hash chain.

Lastly, Achieve advancements for authenticated key agreement protocols constituting a secure and efficient for universal purpose authentication over unreliable adversarial network environments.

Security analysis explains that our solutions are outstanding than existing ones and art suitable for public unreliable networks in respect of security purpose. It is this research that stimulated this dissertation targeted at designing effective AKE protocols well suited for public unreliable network environments as well as meeting strong security.

1.2 Organization

This thesis suggests a broad study on AKE protocols. The main purpose of our study is to give secure and efficient secure solutions to the problem of AKE protocols, they can operate at or below the application-layer (for example, logins to web sites) making it appropriate for network-layer situations (for example, authentication to

wireless networks). In the first part of this dissertation, we concentrate on the security analysis of earlier protocols for AKE to derive the security problems intrinsic in designing AKE protocols. On the base of the prior work of analyzing some existing protocols, we make a proposal of improvements of the security for the earlier AKE protocols. In the next part, we leverage to achieve advancements of secure and efficient authenticated key agreement protocols for adversarial network environments which are unreliable: (1) elaborate authenticated key agreement protocol protecting user privacy, (2) secure and efficient authenticated key agreement protocol protecting user privacy.

We next give an overview our contributions and explain how they are composed in this dissertation. In Chapter 2, we start by emphasizing the characteristics of AKE security demands that we shall satisfy in this dissertation. It includes security purposes, attributes and patterns of attacks.

In Chapter 3, we examine and present cryptanalysis on various related works for AKE which is meeting security threats like AKE protocol, one-time-password AKE protocol and hash-based protocol. We suggest advancements for securing the protocols and resolving the problems on the earlier established AKE protocols. The advanced schemes keep conserving all of the advantageous security characteristics. Our enhancements are based-on the hash-chaining and secret-key and one-time password in Section 3.1-3.3. We give both accuracy and security of our protocols, examine its performance with regard to various cost

parameters, consider design and implementation options, and make a comparison between it and earlier approaches as well.

In Chapter 4, we suggest enhancements of secure and efficient authenticated key agreement protocols fairly optimized for unreliable adversarial network environments. We enhance an elaborate authenticated key exchange protocol conserving user privacy in Section 4.1- 4.2. We suggest a secure-efficiently AKE protocol conserving user privacy. Security analysis explains that our solutions are outstanding than existing ones and are suitable for public unreliable networks in respect of security purpose.

Finally, we discuss whole design schemes that can be obtained and generalized from our research in the whole chapter 4. We also establish the secret key design features ensured success of advanced protocols, and universalize these to develop design schemes that can be used to other kinds of AKE protocols. We sum up the contributions of our research and discuss future plans.

2. Secure Authenticated Key Exchange

To patch existing protocols and plan future protocols, it is necessary to understand how security of AKE protocols fails. In this chapter, we examine and analyze several protocols that aid in the analysis of security protocols in the AKE for session initiation protocol, hash-based AKE protocols and one-time password AKE protocol. The analysis of existing real protocol, device, or design requires the protocol described. To meet the various needs in specific environments. We start by emphasizing the characteristics of AKE security requirements that we will apply in our dissertation. In this dissertation, we demonstrate that it is not important how many protocols are adoptive, as claimed in the problem statement of each section.

2.1 AKE Security

Secure AKE protocol is a basis core element in the secure channel cryptography [46, 47, 48], which two or more parties participate the process sharing a secret key being available for subsequent cryptographic use [45]. Authenticated key exchange (AKE) ensures authenticity of the parties as well as allows parties to calculate the shared key. For achieving secure channel over unsecure public, untrusted available unsecure networks, actually, mutual authentication is essential.

For AKE protocols there are many potential threat scenarios and there is no only security definition. This section states the goals of the security and the common techniques of attacks against AKE protocols used to compare between previously published AKE protocols and our enhanced protocols in this thesis.

2.1.1 The Network and Tasks

People hope that their computing devices are maintaining the privacy of their personal information. So as to do like that, before giving access these devices must be asked for evidence of the user's authority, which is called authentication. Once the evidence the user supplying is verified the device will act on the user and server's behalf. In the general internet setting, the whole system composed of computing devices, linked through communication channels. The communication means itself is affected through a lot of computing devices such as switches, router and so on, but we will be able to overlook the details of this. Each computer has several processes operating in it, which communicate with other processes in the same computer or across the network. All the computers and all the processes running in them, the communication channels, the user inputs and outputs are all together can be defined as the network.

The works that we would like to securely realize are over what is generally found in an internet setting today. Conventionally information

security was confined to security of information transmission. However, it is general to take security of information which is the input for, or the output from, various distributed computational works into consideration in theoretical cryptography. Applications of such works would contain private information retrieval, privacy preserving data mining, electronic commerce, or online voting.

2.1.2 Security Requirements

The guarantees of security will be only available to parties which obey the regulated protocols accurately, and other parties cannot access to internal network. Parties which don't follow the protocol, or whose internals become reachable to others are considered corrupts. Corrupted parties may convince with each other and share their datum. On the communication channels the sent messages are likely to be eavesdropped by the corrupt parties. Messages may be changed by corrupt routers, for example, delivered out of order, or never delivered. Furthermore, there is no way to identify the origin of a delivered message.

In networks, computer and mobile phone security as well, a series of events through which intelligent adversary or a secret nation could use the system in a disapproved way to result damage, for instance by hateful compromising the secrecy, honesty, or availability of the system's information. The literature on information security separates

threat situations into three categories, on the basis of what desired attribute of the data is lost: secrecy, honesty, or availability. These main threat-situations can be spoken as follows:

“Information” is exposed to someone

- Someone is called that should not have access to it.

“Information” is modified

- Modified method is in a manner contrary to policy.

“Authorized users” are prevented from accessing materials

- Which are information or resources.

The works of a protocol are defined by the components. As Boyd’s scheme [46] proposed, “any attack on a protocol is only valid if it violates some property that the protocol was intended to achieve”. Therefore, it is important for protocol architect to establish the desirable that a protocol succeeds. Before reviewing previous solutions in this dissertation, we sum up the security demands that an AKE protocol shall satisfy.

Definition (Implicit-Key Authentication)

Let H be a set of parties who desire to share a secret key by operating a key exchange protocol KEP . Let K_i be the key calculated by a party $U_i \in H$ as an outcome of an execution of protocol KEP . We say that KEP

satisfies implicit key authentication if each $U_i \in H$ is confirmed that no $U_a \notin H$ can learn the key K_i unless aided by deceitful $U_j \in H$ or by any other reliable party.

AKE protocols set a goal of generating a session key between solely expressed user & server who have really communicated in a recent operation. An AKE is described in the following definition [45].

Definition (AKE protocol)

An AKE is a key exchange process which serves implicit mutual key authentication protocol.

In countless applications, communication-channel parties establish secure several multi-sessions. Therefore, implicit AK has to be fulfilled even when multi-parties participating in the protocol are run simultaneously, although all the same time adversaries may read, insert, intercept, delete, modify, delay and replay messages. A protocol accomplishing implicit AK is called an AKE protocol, and is essentially importance in much of contemporary network security & cryptography.

There are extra demands needed for security of key-exchange protocols. These demanded conditions have been studied and proposed

by past published solutions. We generally expect that the adversary controls the communications on the network. The adversary can see, change, remove and fake all messages sent.

<Known Key Security>

Consider that the malicious hateful-adversary can get the session keys in sessions. If it is secure under this assumption, a protocol is called by known key security. This is typically and usually considered as a normal key establishments protocols [47]. Form these attacks a malicious adversary gets some keys using this information for determining new keys.

<Perfect Forward Confidentiality>

An AKE protocol gives complete forward confidentiality if exposure of long-term secret secure key materials could not compromise the confidentiality of the key exchanging sessions from past communications [48]. Without authentication, its property does not apply to key exchange.

2.2 The Known Protocol Attack Types

The list of attacks [45, 47, 49] are referred as below;

<**Eavesdropping Attacks**> The protocol compromises the confidentiality when an adversary takes the information sent from the protocol. But, the adversary isn't able to change or disturbing the information security in any way.

<**Modification Attacks**> The hateful malicious adversary compromises the protocol's integrity, modifies and then transmits the information when it takes the session information sent from the session protocol.

<**Replay Attack**>The replay attack is used together with other attack methods. This attack is easy.

<**Reflection Attack**>This attacking methods is a target to make an attack on a two-way authentication system that employs the identical protocol in challenge-response both directions.

<**Impersonation Attack**>An adversary disguise itself as the identity of user/server of the legitimate parties in an unsecure network.

<**Dictionary Attack**>When an user enters a password and logs on, an adversary can accumulate passwords, and then compare probable password to the group of true encrypted passwords with the true passwords.

<**Man-in-the-middle(MITM) Attack**> If the adversary is monitoring, taking, and supervising the user's communications, then we called that it is apt to occur.

From old times, Bellare and Rogaway [21] and Pointcheval and Rogaway [12] indicated the security of AKE. A lot of AKE protocols have been suggested and many of them were broken subsequently. We examine the schemes and suggest the secure AKE protocols [43, 44, 50, 51].

3. Security on Authenticatedkey Exchange

3.1 Review of the Authenticated key protocol

3.1.1 INTRODUCTION

You need the keys to encrypt or decrypt the data for secure communication. These keys are shorter than the data which is to be secured with the keys. Cryptography is used to protect the communicated data by the parties and to store a number of information encrypted. For example, a public network such as a wireless network, the message is possibly modified, inspected and deleted by a third party. The cryptographic systems rely on the secret of the secret key. In this chapter, we propose an improvement after analyzing the AKE protocol previously, to rectify the problem with the previous AKE protocol. Our system is based on a one-time password, symmetric secret key and hash functions chain. We discuss variety of protocols in AKE for session initiation protocol. Both authentication of mutual communicating parties and the confidentiality of data to be transmitted to the other entity are the fundamental steps for establishing secure communications over a network kind like insecure public network. In the presence of the enemy, secure communication in a unsafe public network poses a problem when it is joined to complete the task in a safe manner for a specific purpose even if parties are advised with the experts. In many cases, these public networks are vulnerable to malicious attacks that cause a lot of leakage of important information for fear of tampered device and remote communications. At the same time, the problem is that the security is used and its importance hostile distributed environment and

exchange, confidential information, to increase giving the current trend in modern system design towards the distributed architecture and minimum of trust assumptions. Understanding how the key exchange protocol that was captured authentication to the needs of computing for many applications could be failed, is the key patch both to design the protocol of existing and future protocols.

To supply security-services in public transmission network, AKE is applied as and basic process to establish secret key which is used for authorize each communicating parties. The authentication process requires any of users to submit the article in the secret as a challenge-response value for verification process. Verification-process is performed by the server to provide security-services using the session keys. With the authentication and key exchange process, network resources may be maintained by authenticating legitimate communication partner. Data integrity and confidential information can be ensured by using a private key that is negotiated for message encryption and authentication. Therefore, key exchange that has been authenticated is directly related to network resources, information confidentiality, and for the integrity of the data that are involved in network security.

In many areas of computing to solve safe communication matters, AKE protocols have been designed, but it is still a developing task to increase security. As the new internet connectable devices and small embedded systems have a non-centralized and distributed architecture, password or secret key security will be threatened. The protocols in AKE are constructed to verify communicating entities and create secret shared keys with mutual-trust. Though the mutual-trust is to guard the communicating parties, AKE steps is required to supply security. We supply the building design blocks needed for protocols in AKE to strength ensecurity power and to reduce cryptographic risk.

We examine the security problems on the AKE for session initiation protocol (SIP), hash-based functions and one-time password [52, 53, 54, 55, 56, 57, 58, 59, 60, 61]. Also, we construct improvements of the previous authenticated key agreement protocols. These contributions are based on symmetric secret key encryption, secure hash function protecting user confidentiality, and efficient secure authenticated key agreement protocol.

3.1.2 Review of Huang-Wei's Scheme

The Session Initiation Protocol (SIP) is a standard useful protocol of the internet engineering task force that starting an interactive client session comprising multi-media elements [8, 9]. SIP is the basis of the method of challenge-response protocol. SIP deals with authentication, encryption and digital signatures problems. It is proposed a new efficient authentication scheme for SIP by Huang-Wei [7]. This method is simple as needed just hash function evaluation of seven for the client and server. However, it cannot prevent attacks from the well known attacks.

Before the protocol is started, it is assumed that the server S and client C shares common information $H(PW)$. The key exchange method that has been designed by Huang-Wei, is described as follows:

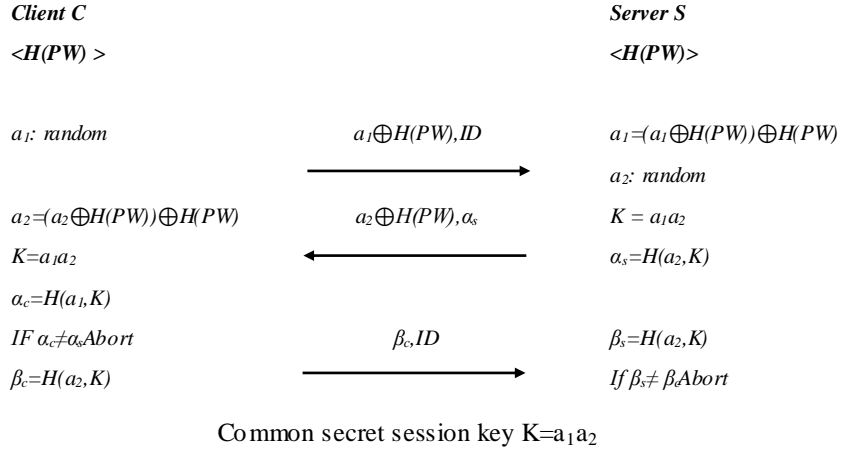


Fig. 3.1 AKE scheme of Huang-Wei's protocol

- (1) $C \rightarrow S: \langle a_1 \oplus H(PW), ID \rangle$. The client C selects a random secret number a_1 , and then sends $a_1 \oplus H(PW)$ and ID to the server S where H is a public one-way-hash function and PW is C's password.
- (2) $S \rightarrow C: \langle a_2 \oplus H(PW), \alpha_s \rangle$. After receiving the $a_2 \oplus H(PW)$ and ID from C, S obtains a_2 by computing $(a_1 \oplus H(PW)) \oplus H(PW)$. Next S selects a random secret number a_2 and computes the session key $K = a_1 a_2$. Then, S sends $a_2 \oplus H(PW)$ and $\alpha_s = H(a_2, K)$ to C.
- (3) $C \rightarrow S: \langle \beta_c, ID \rangle$. C obtains a_2 by computing $(a_2 \oplus H(PW)) \oplus H(PW)$, and then gets $K = a_1 a_2$. Eventually, C computes $\alpha_c = H(a_1, K)$ and verifies that α_c is equal to α_s . If the verification succeeds, then C sends $\beta_c = H(a_2, K)$ to S.
- (4) S authenticates the identity of C using β_c . After receiving β_c from C, S first computes β_s and verifies then verifies that β_s is equal to β_c . If the verification succeeds, S gives C permission to access the resource of S. Moreover, S and C share the common secret session key $K = a_1 a_2$ for securing subsequent communications.

In the method of Huang-Wei, given a rating of 7 hash function above, four exclusive OR operation is performed for the procedure. In addition, the procedure of the scheme is very simple. So, low calculation characteristic simplicity and is very suitable for the scheme for both the authentication server and the client of limited performance hardware.

3.1.3 Security Analysis

We need to point out that the scheme of Huang-Wei can't accomplish its main security goal (for exchanging authenticated key and for the secrecy of password).

<Attack on the Password Security>

Client in the password authentication method, tend to select vulnerable password so that the client can memorize easily. This behavior is very fragile to the attacker who uses a dictionary attack with a guess method and social engineering. The attacker has two attacking tools (off-line/online dictionary attack). Specially, offline dictionary attack is more threaten to the client who has a weakness.

Scheme of Huang-Wei was suffered from off-line dictionary attack. The attack steps are as follows in Fig 3.2:

First steps of Huang-Wei's scheme: a client C sends $(a_1 \oplus H(PW), ID)$ to the server S and In the second step S sends $(a_2 \oplus H(PW), H(\alpha_1, K))$ to C. In this step an attacker A eavesdrop these messages.

Client C

Server S

$\langle H(PW) \rangle$

$\langle H(PW) \rangle$

a_1 : random

$a_1 \oplus H(PW), ID$

$a_1 = (a_1 \oplus H(PW)) \oplus H(PW)$

a_2 : random

$a_2 = (a_2 \oplus H(PW)) \oplus H(PW)$

$a_2 \oplus H(PW), \alpha_s$

$K = a_1 a_2$

$K = a_1 a_2$

$\alpha_s = H(a_2, K)$

$\alpha_c = H(a_1, K)$

IF $\alpha_c \neq \alpha_s$ Abort

$\beta_c = H(a_2, K)$

$c_1 \leftarrow a_1 \oplus H(PW), c_2 \leftarrow a_2 \oplus H(PW), c_3 \leftarrow H(a_1, K)$

Guess PW' ,

$a_1' \leftarrow c_1 \oplus H(PW'), a_2' \leftarrow c_2 \oplus H(PW')$

$K' = a_1' a_2', c_3' = H(a_1', K')$

If $c_3' = c_3$, then $PW' = PW$

Attack \leftarrow Success

Fig. 3.2 Attack scheme for the AKE method of Huang-Wei

Second steps of this scheme: Let $c_1 = a_1 \oplus H(PW)$, $c_2 = a_2 \oplus H(PW)$, $c_3 = H(a_1, K)$. Then A makes a fraud password PW' and compute: $a'_1 = c_1 \oplus H(PW')$, $a'_2 = c_2 \oplus H(PW')$. After this action, the attacker A computes: $K' = a'_1 a'_2$ and $c'_3 = CH(a'_1, K')$. And last steps, IF c'_3 is equal to c_3 , then A gets the correct PW, if not, A repeats the above process until it ends up with correct pw.

<Forward Secrecy>

The forward secrecy is that session key of the previous, are protected from exposure to some information at the moment. Forward secrecy is not provided by Huang-Wei's scheme: immediately long-term password PW has been leaked and the session key of all previous steps are possibly recovered. Failure to achieve the main security goal of AKE in the scheme of Huang-Wei is due to the lack of forward secrecy and offline dictionary attack.

Let

$x_1 = a_1 \oplus H(PW)$ and $x_2 = a_2 \oplus H(PW)$ be the messages transmitted in the target session.

And now, previous session key K with the PW , x_1 and x_2 can be computed as follows:

$$a_1 = x_1 \oplus H(PW') = a_1 \oplus H(PW) \oplus H(PW)$$

$$a_2 = x_2 \oplus H(PW') = a_2 \oplus H(PW) \oplus H(PW)$$

$$K = a_1 a_2$$

Hence, the client's password once published, the attacker can recover access p-word to privileged information conveyed in a previous session.

3.2 General Security Analysis discussion

According to the report at Perdue University, the result of surveys says that almost 3% of the passwords were less than or equal to three characters and 85% of the all passwords were from six to eight characters [10] in length. With eight characters passwords, there are 36^8 possible choices that can be easily solved. Hence, once the attacker has acquired the client's password, he could connect to the server as if he is a legal client. Since Huang's proposed scheme is very simple and needs only seven secure hash-function evaluations between the server and client. We conclude that this scheme is not proper to fulfill and meet the main security mission of an authenticated key exchange protocol. To jump over the vulnerability of the Huang et al.'s scheme for offline dictionary attack, it is required to Yang et al.'s scheme [11] and other well-known password authenticated key exchange protocols [12, 61].

<Hash-based AKE>

In many fields of computing science, the answer to their security needs is still under research with challenges [13]. Researchers have proposed a variety of protocols for authenticated key exchange (AKE) for the purpose of users to be authenticated so that they can access to service providers [14]. The Kerberos is based on the technology of the secret key and symmetric timestamp [15], and is one of AKE protocol that is the most widely used, but has drawbacks of the dictionary attacks, exposure of the session key and replay attack [16]. The improved AKE protocol was proposed to increase the efficiency, security, and scalability of Kerberos [17,18,19,20]. It has been the subject of many researches that the robustness of AKE protocol to the loss of session keys [21, 22, 23]. Bellare and Rogaway have observed that it is essential for the secure authenticated key exchange [21]. They point out that if attacker hold the session key, they can use it only for the session which the key protecting. Especially, it must not be easier for the hackers to calculate other session keys, Chein and Jan [17] state that the security drawbacks in specific session key based on certificate protocols, and then suggested a mixing authentication for big mobile networks which based on public key infrastructure, challenge-response value and hash functions' chaining [24, 25]. But Tang and Mitchell [26] examined the protocol suffered from security defects.

< Problem Statement and Preliminaries >

Nowadays, Shi *et. al.* [27] has proposed an AKE protocol that can be used among user, service provider and key distribution center(KDC). This protocol is depending on symmetric encryption, challenge-response, Diffie-Hellman algorithm and hash function. Shi *et. al.* noted that the proposed protocol is a possibly strong AKE protocol. But if an attacker takes a session key in the protocol scenario, it would be weakened. We research the strength of our improved protocols.

Integrity check of encrypted data prevents a hacker from undetectable tampering with information. A secure hash function is aopenly computed without key function that should not be directly used to supply integrity, that is, an adversary could easily alter a data and derive the new hash. Alternatively, a function with a key, so called a message authentication code(MAC) should be used. The hash-based AKE scheme has utilized a secure hash function to generate the digest of the data. A secure hash function h performs on a variable-length data M generating a hash value $h(M)$ of fixed and short length. Furthermore, a secure hash function h is known as collision-resistant, if it is impossible to get two different messages $x \neq y$ that generates the equal hash value, for example, form a confliction $h(x) \neq h(y)$. For the sake of completeness, we will give a basis of a set of collision-resistant functions. The collision-resistant value $v : N \rightarrow R$ is worthless for all positive polynomial value p and for large enough value k , $v(k) < 1$ over $p(k)$, $\{0, 1\}^*$ stands for the set of strings of the alphabet $\{0, 1\}$. For $x \in \{0, 1\}^*$, $|x|$ identifies the length of x .

Definition (Collision-resistant Hash Function)

Let H be a stochastic algorithm that, on input 1^k , in polynomial time and outputs an algorithm $h : \{0, 1\}^* \rightarrow \{0, 1\}^k$. Then H defines a set of collision-resistant hash function if:

Efficiency For all $h \in H(1^k)$, for all $x \in \{0, 1\}^*$, it takes polynomial time in $k + |x|$ to generate $h(x)$.

Collision-resistance For all families of probabilistic polynomial-time Turing machines A^k , there exists a worthless function $v(k)$ such that

$$\Pr[h \leftarrow H(1^k); (x_1, x_2) \leftarrow A_k(h) : x_1 \neq x_2 \wedge h(x_1) = h(x_2)] = v(k).$$

3.2.1 Review of Shi et al.'s Protocol

Shi *et al.*'s Protocol is made up of two phases: the first and subsequent phase. The KDC(Key Distribution Center) keeps a secret key K_c to generate the secret key for its users. the long-term shared secret key between the KDC and user is K_{uc} , while the long-term shared secret key between the KDC and the server is K_{sc} , let $K_{uc} = f(K_c, U)$, $K_{sc} = f(K_c, S)$, S and U denote their identities, f is a secure hash function. The detailed description is as below.

<First Phase>

II. $U \rightarrow S \langle U, a^x \bmod p, h(a^x \bmod p, K_{uc}) \rangle$

Where p is a large prime and a is a generator with order $p-1$ in $GF(p)$, h denotes a one-way hash function.

1. U randomly selects a secret number x to compute $a^x \bmod p$
2. U calculates $h(a^x \bmod p, K_{uc})$
3. U sends his identity $U, a^x \bmod p, h(a^x \bmod p, K_{uc})$ to server

I2. $S \rightarrow KDC \langle U, a^x \bmod p, h(a^x \bmod p, K_{uc}), S, a^y \bmod p, h(a^y \bmod p, K_{sc}) \rangle$

1. S store $a^x \bmod p$.
2. S randomly selects a secret number y to compute $a^y \bmod p$.
3. S calculates $h(a^y \bmod p, K_{sc})$.
4. S sends $U, a^x \bmod p, h(a^x \bmod p, K_{uc})$ and his identity $S, a^y \bmod p, h(a^y \bmod p, K_{sc})$ to KDC .

I3. $KDC \rightarrow S \langle E_{K_{sc}}(a^y \bmod p, n, r_n), E_{K_{uc}}(a^x \bmod p, a^y \bmod p, n, r_n) \rangle$

1. KDC authenticates U and S by checking the hash value

$$h(a^x \bmod p, K_{uc}) = h(a^x \bmod p, K_{uc})' \text{ and}$$

$$h(a^y \bmod p, K_{sc}) = h(a^y \bmod p, K_{sc})' \text{ separately}$$

2. KDC choose n and generates a random number r_n , where n is the number of times that U can communicate with S
3. KDC sends $E_{K_{sc}}(a^y \bmod p, n, r_n), E_{K_{uc}}(a^x, a^y \bmod p, n, r_n)$ to S .

I4. $S \rightarrow U \langle E_{K_{uc}}(a^x \bmod p, a^y \bmod p, n, r_n), E_{K_n}(a^x \parallel a^y \bmod p) \rangle$

1. S decrypts message $E_{K_{sc}}(a^y \bmod p, n, r_n, E_{K_{uc}}(a^x \bmod p, n, r_n, a^y \bmod p, n, r_n))$ using K_{sc} .
2. S authenticates KDC by checking the value of $a^y \bmod p$
3. S computes $a^{xy} \bmod p$ to get K_n and encrypts $a^x \bmod p, a^y \bmod p$ using K_n .
4. S sends $E_{K_{uc}}(a^x \bmod p, a^y \bmod p, n, r_n), E_{K_n}(a^x \parallel a^y \bmod p)$ to U
5. S keeps $\langle U, K_n, n, r_n \rangle$

I5. U receives $\langle E_{K_{uc}}(a^x, a^y \bmod p, n, r_n), E_{K_n}(a^x \parallel a^y \bmod p) \rangle$

1. U decrypts message $E_{K_{uc}}(a^x, a^y \bmod p, n, r_n)$ using K_{uc}
2. U authenticates KDC by checking the value of $a^y \bmod p$
3. U computes the $a^{xy} \bmod p$ to get K_n
4. U decrypts $(a^x \bmod p \parallel a^y \bmod p)$ using K_n
5. U authenticates S by checking the value of a^y and $a^x \bmod p$
6. U keeps $\langle K_n, n, r_n \rangle$

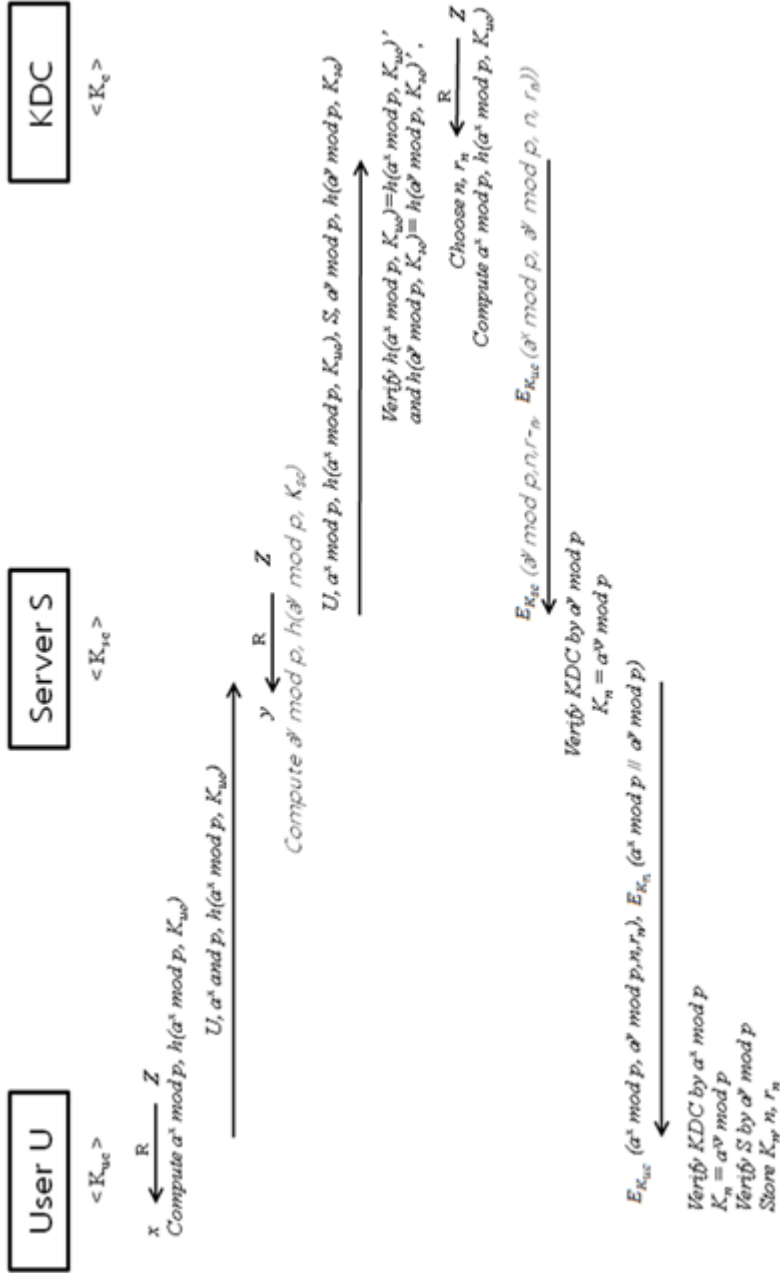


Fig. 3 The First phase of Shi et al.'s protocol

<Next Phase>

In the subsequent step, the user is requesting the **i-th** job now.

S1. $U \rightarrow S \langle U, E_{K_{n-i+1}}(r_{n-i+1}) \rangle$

1. **U encrypts r_{n-i+1} using K_{n-i+1}**
2. **U sends $\langle U, E_{K_{n-i+1}}(r_{n-i+1}) \rangle$ to S.**

S2. $S \rightarrow U \langle E_{K_{n-i}}(r_{n-i+1} \| r_{n-i}) \rangle$

1. **S decrypts the message $E_{K_{n-i+1}}(r_{n-i+1})$ using K_{n-i+1}**
2. **S authenticates U by checking $r_{n-i+1} \neq r'_{n-i+1}$**
3. **If $r_{n-i+1} = r'_{n-i+1}$, S computes $K_{n-i} = h(K_{n-i+1} \| r_{n-i+1})$**
4. **S generates a new random number r_{n-i} .**
5. **S encrypts $(r_{n-i+1} \| r_{n-i})$ using K_{n-i+1}**
6. **S stores K_{n-i} and r_{n-i} , updates i .**

S3. U receives $\langle E_{K_{n-i}}(r_{n-i+1} \| r_{n-i}) \rangle$

1. **U computes $K_{n-i} = h(K_{n-i+1} \| r_{n-i+1})$**
2. **U decrypts $E_{K_{n-i}}(r_{n-i+1} \| r_{n-i})$, and authenticates U by checking $r_{n-i+1} = r'_{n-i+1}$**
3. **If $r_{n-i+1} = r'_{n-i+1}$, U stores K_{n-i} and r_{n-i} updates i .**

3.2.2 Shi et al.'s Protocol weakness & security

This part, we will show that Shi et al.'s Protocol struggles for a known key attack. Therefore we think about a known key attack, we anticipate that the session key K_{n-i+1} is uncovered to an attacker A.

The attacking scenario is described in Fig4. The attack progresses as follows:

1. When a user U is requesting the i -th service in the subsequent phase, the attacker A eavesdrops the transmitted message $\langle U, E_{K_{n-i+1}}(r_{n-i+1}) \rangle$ and $\langle E_{K_{n-i}}(r_{n-i+1} || r_{n-i}) \rangle$. With the message $\langle U, E_{K_{n-i+1}}(r_{n-i+1}) \rangle$, attacker A can obtain r_{n-i+1} by decrypting $E_{K_{n-i+1}}(r_{n-i+1})$.
2. Having K_{n-i+1} and r_{n-i+1} , A is also to compute the next key $K_{n-i} = h(K_{n-i+1} || r_{n-i+1})$. By decrypting $E_{K_{n-i}}(r_{n-i+1} || r_{n-i})$ in the message $\langle E_{K_{n-i}}(r_{n-i+1} || r_{n-i}) \rangle$, A obtains r_{n-i} . Then A records $K_{n-i} = h(K_{n-i+1} || r_{n-i+1})$ and r_{n-i} for a new session.

Thus the server and attacker ha the same key which has been shared between the server and the user in Shi et al.'s Protocol. It means that this protocol cannot be proven safe security models for key establishment protocol [21]. In the Shi et al.'s protocol it is claimed to have infringed the old session key K_{n-i+1} which is shared by the server and the user, the known key attack still fails. However, we will show that the claims are provable security for this protocol was wrong.

3.2.3 Our improvement on This Protocol

In order to provide security over the known key attack, we suggest couple of advanced AKE protocols that enable the parties establishing a session key that could be applied to make their subsequent communications secure. However this known key attack is complicated and unlikely, it is possible to avoid it.

3.2.4 Improvement basis on the Symmetric key & hash chain

We have examined to in previous section, the hacker gets a session key and then utilize this information to compute new session keys. It is certain that whether both U and S use different shared secret value in S1 of the next stage, the attacker could not be possible to calculate r_{n-i+1} and new session keys by taking lessons a session key and sniffed communicating messages. Hence, both U and S must have different secret value for protecting the session from the well known key attack.

For the sake of keeping the protocol secure, KDC computes long-term shared key $K_{us} = f(K_c, U, S)$ between the server and the user.

After executing the first phase of this protocol, S and U process the following steps to demand the i -th service in the next phase. First is as follows:

S1. $U \rightarrow S \langle U, E_{K_{n-i+1}} (E_{K_{us}} (r_{n-i+1})) \rangle$

1. U encrypts r_{n-i+1} using K_{us} and K_{n-i+1} .
2. U sends $\langle U, E_{K_{n-i+1}} (E_{K_{us}} (r_{n-i+1})) \rangle$ to S.

S2. $S \rightarrow U \langle E_{K_{n-i}} (r_{n-i+1} \| r_{n-i}) \rangle$

1. S decrypts the message $E_{K_{n-i+1}} (E_{K_{us}} (r_{n-i+1}))$ using K_{n-i+1} and K_{us}
2. S authenticates U by checking $r_{n-i+1} \neq r'_{n-i+1}$
3. If $r_{n-i+1} = r'_{n-i+1}$, S computes $K_{n-i} = h(K_{n-i+1} \| r_{n-i+1})$
4. S generates a new random number r_{n-i} .
5. S encrypts $(r_{n-i+1} \| r_{n-i})$ using K_{n-i+1}
6. S stores K_{n-i} and r_{n-i} , updates i .

S3. U receives $\langle E_{K_{n-i}} (r_{n-i+1} \| r_{n-i}) \rangle$

1. U computes $K_{n-i} = h(K_{n-i+1} \| r_{n-i+1})$
2. U decrypts $E_{K_{n-i}} (r_{n-i+1} \| r_{n-i})$, and authenticates U by checking $r_{n-i+1} = r'_{n-i+1}$
3. If $r_{n-i+1} = r'_{n-i+1}$, U stores K_{n-i} and r_{n-i} updates i .

We ameliorate Shi et al.'s Protocol in the basis of the merit of hash chain and secret key cryptosystem [28]. In this ameliorated protocol, the long-period shared key K_{uc} and K_{sc} are identical to those of Shi et al.'s Protocol. The first phase of betterment protocol is also same as Shi et al.'s Protocol. In the next state, they anticipate the user U is call for the i -th service now. The next stage of ameliorated protocol is as follows:

S1. $U \rightarrow S \langle U, E_{K_{n-i+1}}(h^{n-i+1}(r_{n-i+1})) \rangle$

1. U computes $(h^{n-i+1}(r_{n-i+1}))$
2. U sends $\langle U, E_{K_{n-i+1}}(h^{n-i+1}(r_{n-i+1})) \rangle$ to S.

S2. $S \rightarrow U \langle E_{K_{n-i+1}}(h^{n-i}(r_{n-i+1} \| r_{n-i})) \rangle$

1. S decrypts the message $E_{K_{n-i+1}}(h^{n-i+1}(r_{n-i+1}))$ using K_{n-i+1}
2. S checks whether $h^{n-i+1}(r_{n-i+1})$ equals the previous hash value $h^{n-i+1}(r_{n-i+1})$.
3. If the check value is good, S generates a new random number r_{n-i} and computes $K_{n-i} = h(K_{n-i+1} \| r_{n-i+1})$.
4. S encrypts $(h^{n-i}(r_{n-i+1} \| r_{n-i}))$ using K_{n-i+1}
5. S stores K_{n-i} and r_{n-i} , updates i .

S3. U receives $\langle E_{K_{n-i}}(h^{n-i}(r_{n-i+1} \| r_{n-i})) \rangle$

1. U computes $K_{n-i} = h(K_{n-i+1} \| r_{n-i+1})$
2. U decrypts $E_{K_{n-i}}(h^{n-i}(r_{n-i+1} \| r_{n-i}))$, and checks whether $h^{n-i}(r_{n-i+1})$ equals the previous hash value $h^{n-i}(r_{n-i+1})$.
3. If the check succeeds, U stores K_{n-i} and r_{n-i} updates i .

3.2.5 Security Analysis

To protect the attacks, we should prove how to block information possessed in a few sessions from the information to be used for a winning attack on any different session. Custom answer is that the easiest way to achieve this is using a symmetric key which is shared only to S and U. Our method of settlement is to construct the AKE protocol which deals authenticated key agreement protocol with password that allows establishing of a robust session key from a likely weak password. In the preceding section, both S and U employ a long-period shared key K_{us} for sake of protecting communications in the succeeding phase.

Expect that an hacker A learns a session key K_{n-i+1} and tapped messages $\langle S1, S2 \rangle$. Without knowing the long-period shared key K_{us} , A cannot extract r_{n-i+1} and compute K_{n-i} in the altered protocol. But, only the legal user can calculate the session key.

Thereby the hash function has one-way property, even though a session key K_{n-i+1} is compromised, the hacker also need to know the token $h^{n-i}(r_{n-i+1})$ to calculate K_{n-i} . Hence, the attacker cannot calculate the $h^{n-i}(r_{n-i+1})$. Therefore, attack is no longer effective against the meliorated protocol. Suggested protocol has the property that it is the novel way of mixing an asymmetric hash chain and a symmetric key. Suggesting protocol supplies the known key security purpose. Because the hash chain produces an asymmetric setting, thus an enemy's compromising the server to capture the authentication secret cannot accomplished in impersonating a legal user at the succeeding login.

3.2.6 Conclusion

We inspect the security of two previous methods for authenticated key exchange (AKE) protocol which meets the security menace.

First of all, we treat the matters such as forward secrecy and an offline dictionary attack.

And next, we have demonstrated security flaws of Shi et al.'s protocol. To cure this problem, we suggested improvements to the basic protocol so that the protocol can be more secure.

4. Authenticated key Exchange Protocols

Key control refers to how keys are provided to and updated by each entity in a protocol. It is notoriously complex problem since spreads keys often needs a secure channel. Authentication is the steps by which parties demonstrate their identity in a protocol. Only one party to be authenticated is often insufficient: in cases of such bi-directional authentication is requested. There exist some problems that could happen when one entity is authenticated. These consider attack of the man-in-the-middle while a hacker pretends to be entity C to S and entity S to entity C, interceding and probably altering all communication between them. While one can construct safe authentication protocols for the purpose of smart cards, if it is true, in the protocol, simply using smart cards does not mean that the protocol is safe enough [53. 54. 55. 56.57, 58, 59, 60, 61]. The authentication protocol should employ the smart cards in a security-considered and intelligent way. Unluckily, a lot of preceding protocols provides not enough additional security while using smart cards and, in fact, reveals the protocol to some attacks. This chapter, we suggest a reformed scheme to preserve user anonymity and fix weaknesses of Liao-Chen-Wang (LCW) scheme in Section 2.1. Furthermore, we suggest an ameliorated scheme to supply user privacy over both outside adversaries and a remote server in Section 2.2. Client confidentiality is clearly an additional required feature. In many areas this adjunctive property could be really essential matter, since confidentiality is frequently an important requirement. It is very valuable to design efficient and practical privacy preserving authentication skills.

4.1 The Improved AKE

Authenticated key exchange is a crucial process to set up a secure transmission over public unsecured networks. After that Bellovin and Merritt announced proposal of a pw-based AKE protocol secure against dictionary attacks [29], many researchers has proposed PAKE protocols [30,31,32,33]. PAKE protocols need a client to memorize its short length password and make participating entities hold a session key in common and have authenticated each entity. Otherwise, because the password length is short, extra ordinary assist should be taken while constructing prototypes so making sure that each two key and the password managed that remain secret in the end. A shared password authentication scheme is usually adopted to supply authenticating a remote server and legitimate users to the others over public unsecured networks. After that Lamport [35] has introduced a password-based remote authentication method, many scientists made proposals on the authentication schemes to advance security and efficiency, just like as EKE[29] and Password-based Authenticated Key Exchange(PAKE)[30, 34]. The password is a really easy way to authenticate. In terms of portability and simplicity they are very complex to match. Although the prototype and security point of authentication based on password methods have a long period of history, all of them has an inherent flaws. Smart card has been broadly applied in many districts of cryptographic protocols because of their low cost, cryptographic capabilities and portability. Pw-based authentication methods also utilized a smart card for sake of providing safe token for even more advanced computation. However, it restricts the necessary elements in smart cards; the calculation and the transmission cost should be cheap for field appliances. In this chapter, we propose an advanced design to terminate these flaws and keep user

anonymous, a fundamental issue in electronic commerce programs.

4.1.1 Hwang-Lee-Tang Scheme's Problem

Hwang-Lee-Tang has made a proposal of authentication scheme between a remote user and server [36] in 2002. However the proposal could authenticate a client, bidirectional authentication among user and server failed and key agreement of the session. The proposal couldn't be freed from the problem of time sync. In early 2003, Chien-Jan has made proposal of authentication scheme that is based on nonce value and a smart card [17]. The proposal supplies for the client and the server mutual authentication from each other. Although, it is needed to make up a confirm chart in the proposal. A legal user couldn't up to date the users password efficiently and easily while it confront the potential security threats. Juang made a proposal of a authentication protocol for the password that gives a secret key management function based on smart cards [37] in 2004. We analyzed the processes in the protocol and its proof of security. We have proposed an advanced protocol to terminate these flaws and conserve user anonymity. And in this chapter, we will go with reviewing Liao et al.'s (LCW) protocol, and next, point out the details [39] of weaknesses in the protocol and its security work. We also explain our advanced protocol and examine our protocol's security.

4.1.2 Reviews of Liao-Chen-Wang's Protocol

The every phase of registration, login, authentication key agreement, and password update in their protocol is described follows. Figure 4.1 demonstrates Liao et al.'s protocol.

US_u : U 's registration time

HF : A secure hash-function.

$TUID_u$: U 's transformed-identity

ID_u : The user U 's identity

PW_u : The user U 's password

b : The permanent secret key of server S .

p : A large prime number

g : A primitive-element in Galois field $GF(p)$.

r_u and r_s : Random numbers generated by user U and server S , respectively.

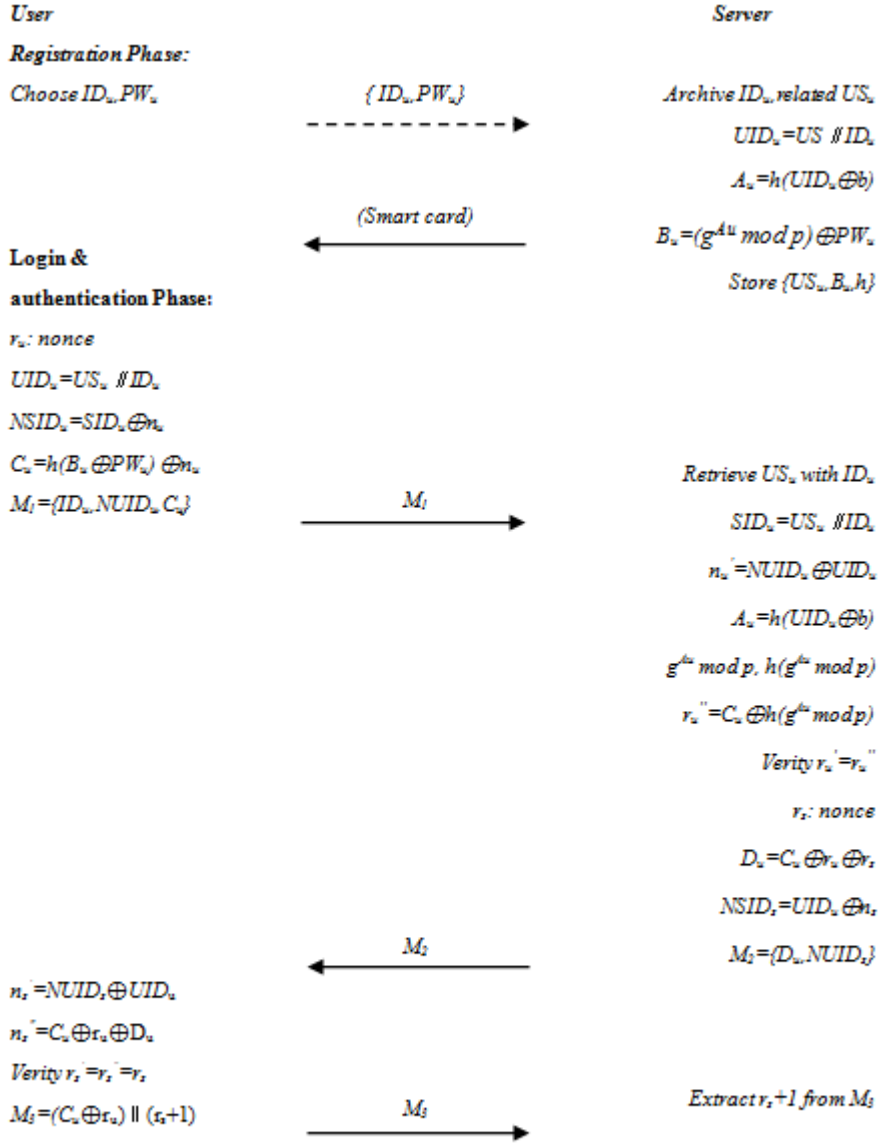


Figure 4.1: LCW-Scheme

<Phase of registration>

This step is invoked one time when user U registers to S in initialization.

1. U send $\langle ID_u, PW_u \rangle$ to S .
2. S acquires US_u and archives ID_u . Then, S computes $UID_u = US_u || ID_u$, $A_u = h(SID_u \oplus x)$, $B_u = (g^{A_u} \bmod p) \oplus PW_u$. Next S stores UID_u, B_u and h in a smart card and issues the smart card to U .

<Phase of Login & Authentication>

When U intends to login S , U and S share the information mutually authenticate each other

1. U connects his smart card to a reader device. For U 's ID_u and PW_u the smart card challenges U , then generates and stores a nonce r_u . Next, to generate $UID_u = US_u || ID_u$, retrieve US_u and compute $NUID_u = UID_u \oplus n_u$ and $C_u = h(B_u \oplus PW_u) \oplus n_u$. Finally, U sends the $M_j = \{ID_u, NUID_u, C_u\}$ to S .

2. S retrieves US_u after receiving the message M_1 . S computes $SID_u = US_u || ID_u$, $n'_u = NUID_u \oplus UID_u$, $A_u = h(UID_u \oplus x, g^{Au} \bmod p)$, then $h(g^{Au} \bmod p)$, $r''_u = C_u \oplus h(g^{Au} \bmod p)$. If $r'_u = r''_u$, the received $NUID_u$ is truly sent from U and $r'_u = r''_u = r_u$. Hence, U is authenticated. S stores r_u . Otherwise, S terminates the connection. S creates a nonce r_s , computes $D_u = C_u \oplus r_u \oplus r_s$ and $NUID_u = UID_u \oplus r_u$. Then S sends the message $M_2 = \{D_u, NUID_u\}$ to U .
3. U computes $r'_s = NUID_u \oplus UID_u$ and $n''_s = C_u \oplus r_u \oplus D_u$ after receiving the message M_2 . If $r'_s = r''_s = r_s$, S is authenticated. U keeps r_s and computes $M_3 = (C_u \oplus r_u) || (r_s + 1)$. Then, U sends the message M_3 to S . The parameter $r_s + 1$ is the response to S .
4. Since $B_u \oplus PW_u = g^{Au} \bmod p$, $C_u = h(B_u \oplus PW_u) \oplus n_u = h(g^{Au} \bmod p) \oplus r_u$. Thus, $C_u \oplus r_u = h(g^{Au} \bmod p)$. So, $M_3 = (C_u \oplus r_u) || (r_s + 1) = h(g^{Au} \bmod p) || (r_s + 1)$. S can easily extract $r_s + 1$ from M_3 and find r_s in there. At this time, S ensures that U has the nonce r_s .

<The steps of key agreement>

Subsequently receiving r_s that is sent from S , U produces a session key $UK_u = h((B_u \oplus PW_u) || r_s || r_u)$. One time S guarantees that U holds r_s , it **computes** a session key $UK_s = h((g^{Au} \bmod p) || r_s || r_u)$. While $B_u =$

$(g^{A_u} \bmod p) \oplus PW_u$, phase of key agreement is arrived and the key for the session transfer is

$$SK_u = SK_s = h((B_u \oplus PW_u) \parallel n_s \parallel n_u) = h((g^{A_u} \bmod p) \parallel n_s \parallel n_u).$$

<The steps of password update>

Since the user required to change his password, user adds the smart card into a reading device, broadcasts a password change demand at user's machine and keys PW_u . And next, the smart card computes $B_u \oplus PW_u$ and U sends a new password PW_u^* . At the end, the smart card computes $B_u^* = (B_u \oplus PW_u) \oplus PW_u^*$ and substitutes B_u with new derived B_u^* .

LCW protocol applied a UID to keep away from identity duplication. But, despite of a protocol's security, LCW protocol is insecure with the existence of an live opponent. Following section represents that an opponent may attack UID to success offline dictionary attack. The protocol is also revealed to MITM attacks. Their asserting theorems and proofs are not correct.

4.1.3 Drawbacks on this Scheme

One time a smart card is captured by an enemy, all the stored information in it is disclosed to the enemy. Following vulnerable

points exists in smart card [62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80]:

- (1) To compute necessary operations in the protocol it has memory with a microprocessor embedding.
- (2) It can be possible to extract information which is stored in the device (smart card) by checking the power status [40, 41].
- (3) And also reverse engineering skills can be applied to extracting clues from the smart card.

<MITM attack>

It probably happens that an attacker *A* disturbs the transmission between *S* and *U*. The scenario of attacking is summarized in Fig 4.2, Following is more complicated explanation of the attack:

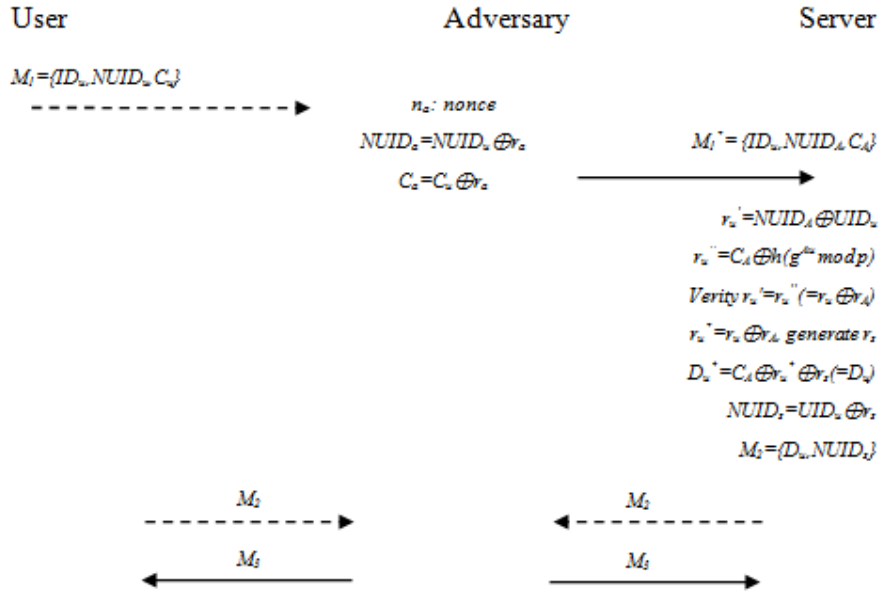


Figure 4.2 Man-in-the-middle attack on LCW scheme

1. In the login authentication phase, when U sends the message $M_1 = \{ID_u, NUID_u, C_u\}$ to S , A intercepts the message M_1 .
2. Using intercepted message M_1 and creating a nonce r_A , A computes $NUID_A = NUID_u \oplus r_A$ and $C_A = C_u \oplus r_A$, A forges a message : $M_1^* = \{ID_u, NUID_A, C_A\}$. Then, A sends M_1^* to S .

3. Upon receiving the message M_1^* , S retrieves TS_u , corresponding to ID_u , and computes $UID_u = US_u \parallel ID_u$, $r'_u = NUID_A \oplus UID_u$, $A_u = h(UID_u \oplus x)$, $g^{Au} \bmod p$, then $h(g^{Au} \bmod p)$ and computes $r''_u = C_A \oplus h(g^{Au} \bmod p)$. Since $r'_u = r''_u$, the communication continues. In this situation, both r'_u and r''_u are equal to $r_u \oplus r_A$ where r_A is generated by A . Thus the forged message passes the verification test of S , S thinks U is authenticated. Furthermore, S keeps $r_u^* = r_u \oplus r_A$ at the server and generates a nonce r_s and computes $D_u^* = C_u \oplus r_u^* \oplus r_s$, $NUID_s = UID_u \oplus n_u$. Actually, $D_u^* = D_u = C_u \oplus r_u \oplus r_s$. Then, S sends the message $M_2 = \{D_u, NUID_s\}$ to U . However, this message is intercepted by A .
4. A forwards the message M_2 to U . Then U operates, as specified in
5. LCW scheme, and sends the message $M_3 = (C_u \oplus r_u) \parallel (r_s + 1)$ to S . however, this message is intercepted by A and forwards this message to S , as if it originated from U .
6. According to LCW scheme, upon receiving the message M_2 , S computes $B_u \oplus PW_u = g^{Au} \bmod p$, $C_A = h(B_u \oplus PW_u) \oplus r_u^* = h(g^{Au} \bmod p) \oplus r_u^*$. Thus $C_A \oplus r_u^* = h(g^{Au} \bmod p)$. Therefore, $M_3 = h(g^{Au} \bmod p) \parallel (r_s + 1)$. Since M_3 is valid, this passes, verifying U has r_s .

7. Following this, S will compute the wrong session key $SK_s = h((g^{Au} \bmod p) \parallel r_s \parallel r_u \oplus r_A)$. However, S can't detect the generation of this wrong session key, because S authenticates U by verifying r_u^* ($= r_u \oplus r_A$). From now, U and S shall use mutually different session keys in encrypting/decrypting their messages. Unlike LCW security analysis, the forged messages made by A pass the verification test of U and S , because communicating parties check the validity of the received messages using the nonce. LCW scheme can't detect this attack and prevent communicating parties from maintaining the invalid sessions. Through this attack, A can make two parties believe and use an unintended session key.

<Attack against Password Security>

LCW protocol does not promise its essential purpose of password security. We will illustrate this point by verifying LCW protocol is weak to dictionary hacking in off-line. Let's set U 's smart card is endangered by the enemy A . Afterward A gathers all the knowledge $\langle US_u, B_u, h \rangle$ that is saved in smart card. Once A calculates transmitted messages among U and S , A is able to execute the next off-line dictionary hacking directly in absence of interacting with S .

1. Using eavesdropped and stored session message

$M_1 = \{ID_u, NUID_u, C_u\}$, A can calculate $UID_u = US_u \parallel ID_u$ and obtains n_u by computing $n_u = NUID_u \oplus UID_u$, and computes $K = C_u \oplus r_u = (B_u \oplus PW_u)$.

2. Then A can perform an off-line password guessing attack to obtain PW_u by guessing a candidate password PW_u' and computing $K = (B_u \oplus PW_u')$. If $K' = K$, which implies $PW_u' = PW_u$, A has successfully guessed U 's password. Otherwise, A tries another candidate password.

In LCW scheme, they introduced and adopted the transformed identity $UID_u = US_u || ID_u$ to avoid identity duplication. However, A may exploit UID_u to achieve the offline guessing attack. Therefore, unlike the authors' claim, without knowing b or A_u , A can impersonate the legal user U freely using the above attack.

<Other Drawbacks>

1. For the sake of proving the robustness of interactive authentication, these steps are supplied with their inspection in the authentication and login steps of LCW protocol [38]. First, Liao's scheme assert as follows: if $r'_u = r''_u$, and then U is authenticated. Nevertheless, as previously spoke in our MITM hacking. A may intercept M_1 and re-create a dialog M_1^* and forward S with M_1^* . Hence, different from Liao's scheme by theorem 1, proof of $r'_u = r''_u$ does not proof $NTID_u$ that is really transferred by U . Likewise the verification of theorem2 doesn't imply S is authenticated, too. Hence, from the.1 and the.2, the exactness in the interactive mutual authentication between S and U can not verified. As discovered in our MITM assault, interactive authentication isn't reached in LCW protocol.

2. In LCW protocol, while enemy A obtains transferred messages between S and U , the adversary A could know who made conversation with S . Now a days, the authentication protocol is not only concentrated on supplying interactive key exchange authentication, but also keep reserving anonymity of user, it is because of user credential is very important point in various e-commerce appliances. LCW protocol is also weak to attack that is occurred inside the user land. S could launch an attack inside obviously, but this is never desirable, while in the registration-step, S transmits PW_u to S .

This job has regarded as LCW key agreement authentication security. Nevertheless Liao et al. asserted verification of its security, the protocol is not secure over an off-line dictionary hacking and MITM hacking and find vulnerabilities in the verification of the proof. Hence, the unsuccess of LCW protocol to accomplish AKE security is solved using MITM assail and off-line dictionary hacking on the scheme.

4.1.4 The Procedures of Our Improved Scheme

In this section, we propose an improved authenticated key agreement scheme that resolves the security drawbacks described in the previous section. Fig 4.3 illustrates the high level approach of our proposed protocol. Let E_k and D_k denote the symmetric encryption/decryption function using symmetric key k satisfying $D_k(E_k(m))=m$. Let SK_u and SK_s are the session keys generated by U and S , respectively. If the scheme ends successfully, then $SK_u=SK_s$.

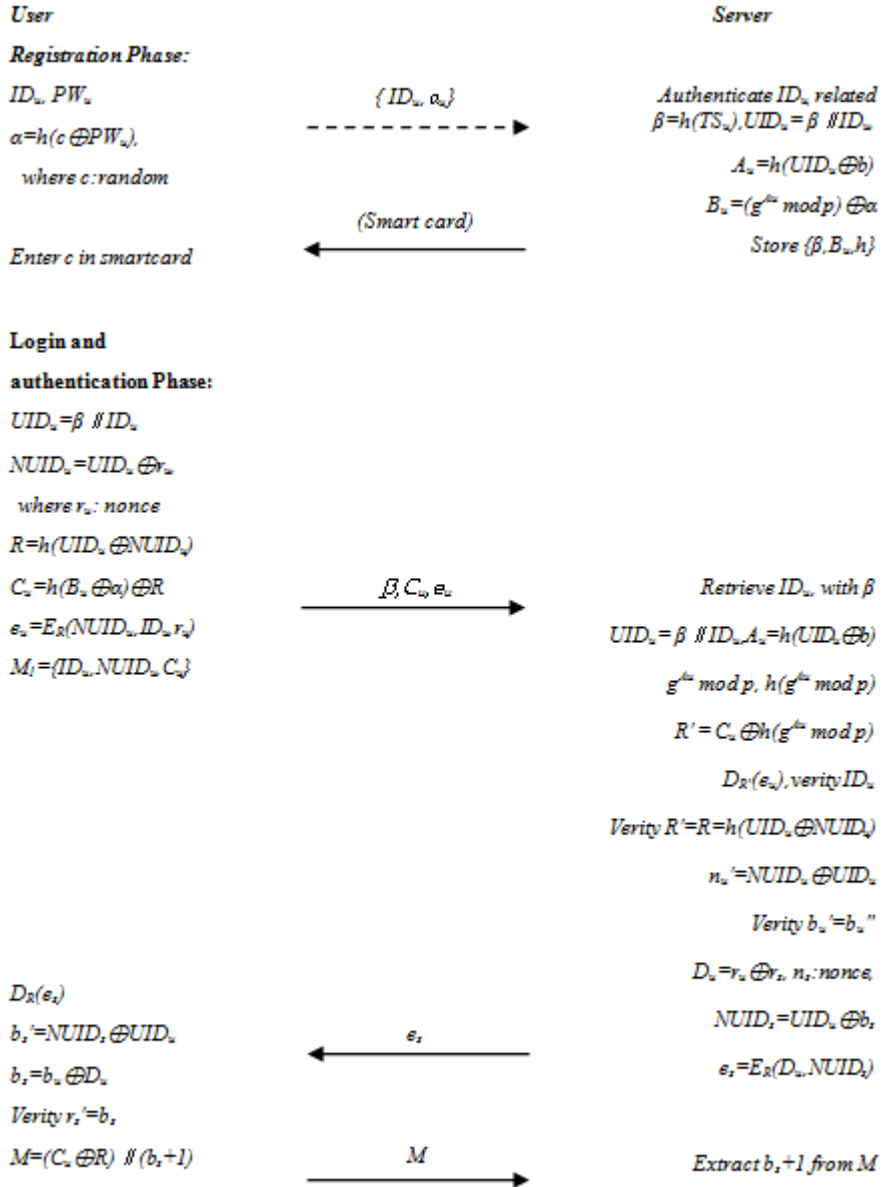


Figure 4.3: Proposed AKE-protocol

<Registration Phase>

The phase of registration is appealone time, while U firstly record to S , and is described as below:

1. U chooses ID_u and PW_u , generates a random number c , then computes $\alpha = h(c \oplus PW_u)$ and submits the registration request $\langle ID_u, \alpha \rangle$ to S via a secure communication channel.
2. Upon receiving the registration request, S acquires the registration time TS_u and archives U 's ID_u and related $\beta = h(US_u)$ for later use. Then S computes $UID_u = \beta || ID_u$, $A_u = h(TID_u \oplus b)$ and $B_u = (g^{A_u} \bmod p) \oplus \alpha$. Finally, stores the values β , B_u and h in a smart card and issues the smart card to U .
3. U enters b into the smart card, then U 's smart card contains β , B_u , h and b . From now on U does not need to remember c .

<Login and Mutual-Authentication Steps>

Next step is invoked whenever user U seeks to login server S .

1. U connects his smart card to a reader. The smart card challenges U for ID_u and PW_u , which are selected at U 's application. Then the smart card generates a nonce n_u and retrieves the stored β to generate the transformed identity $TID_u = \beta || ID_u$. Next, U 's smart card computes $NTID_u = TID_u \oplus n_u$, $R = h(TID_u \oplus NTID_u)$ and $C_u = h(B_u \oplus \alpha) \oplus R$. Then, U encrypts $NTID_u$, ID_u and n_u using R , yielding $e_u = E_R(NTID_u, ID_u, n_u)$. Finally, U sends (β, C_u, e_u) to S .
2. Upon receiving U 's login request, S retrieves ID_u corresponding to β . If no such ID_u matches, S disconnects the request. Otherwise, S computes $TID_u = \beta || ID_u$, $A_u = h(TID_u \oplus x)$, $g^{A_u} \bmod p$, $h(g^{A_u} \bmod p)$ and $R = C_u \oplus h(g^{A_u} \bmod p)$. Then, S gets $NTID_u$, ID_u and n_u by decrypting $D_R(e_u)$, and verifies ID_u and $R' = R = h(TID_u \oplus NTID_u)$. Next, S computes $n'_u = NTID_u \oplus TID_u$. If $n'_u = n_u$, The received $NTID_u$ is truly sent from U . Hence, U is authenticated. S stores n_u . Otherwise, S disconnects the connection. S creates a nonce n_s randomly and computes $D_u = n_u \oplus n_s$ and $NTID_s = TID_u \oplus n_s$. Then, S encrypts D_u and $NTID_s$ using R , yielding $e_s = E_R(NTID_s, ID_u)$. Finally, S sends e_s to U .

3. After receiving e_s , U gets D_u and $NTID_u$ by decrypting $D_R(e_s)$. U computes $n'_s = NTID_s \oplus TID_u$ and $n_s = D_u \oplus n_u$. If $n'_s = n_s$, S is authenticated. U keeps n_s at U 's terminal. Otherwise, U disconnects the connection. Next, U computes $M = (C_u \oplus R) \parallel (n_s + 1)$. Then, U sends M to S . the parameter $n_s + 1$ is the response to S .
4. Upon receiving M , since $C_u \oplus R = h(g^{Au} \text{ mod } p)$, $(C_u \oplus R) \parallel (n_s + 1) = h(g^{Au} \text{ mod } p) \parallel (n_s + 1)$. S can easily extract $(n_s + 1)$ from M and find n_s in there. At this time, S ensures that U has the nonce n_s .

<The key agreement Phase>

U calculates a key of session $SK_u = h((B_u \oplus \alpha) \parallel n_s \parallel n_u)$, after getting n_s of nonce value sent from S . If S make sure that the nonce n_s is obtained by U . it computes a key of session $SK_s = h((g^{Au} \text{ mod } p) \parallel n_s \parallel n_u)$. While $B_u = (g^{Au} \text{ mod } p) \oplus \alpha$ is calculated in the steps of registration, the key-agreement is arrived and the session-key for the communication session is

$$SK_u = SK_s = h((B_u \oplus \alpha) \parallel n_s \parallel n_u) = h((g^{Au} \text{ mod } p) \parallel n_s \parallel n_u)$$

<Pw update Steps>

Whenever user U aims to password change, U puts in users device (smart card) into a reading device, states a credential update request to terminal on the side of U and the key PW_u . Then, it computes $B_u \oplus h(b \oplus PW_u)$ and user U sends a updated values PW_u^* . Eventually, it computes $B_u^* = (B_u \oplus h(b \oplus PW_u)) \oplus PW_u \oplus h(b \oplus PW_u^*)$ and substitutes B_u with new B_u^* .

4.1.5 Security Analysis

Password based authenticated key exchange (PKAE) protocols[29, 32], do not need encrypted channels to take care of the password and therefore the added benefit of building a interactive authenticated key of session that is able to be used to preserve a following session. Today's PAKE schemes do not supply confidential protection to client's identity. Dislike other PAKE schemes, our scheme can supply privacy protection to the client's identity with/without the outer encrypted channel. It half-hearted attacks and advances detections, by among servers and users, of MITM hackings. In this section, we simply illustrate that our proposed protocol provides AKE security by drawing resistance of offline dictionary hacking, MITM hacking, lost smart card device attack, and inner side attacks[62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80]:.

1. **Offline dictionary attack for Resistance.** Imagine an enemy A knows or intercepts all the values (β, B_u, h, b) in U 's smart card and intercepts $(\beta, C_u, e_u, e_s, M)$ transmitted between S and U . Even though A utilizes all the sniffed messages and extracted data in U 's smart card, the off-line dictionary attack is unavailable, though A couldn't get R with unknowing $g^{Au} \bmod p$. So that, the advanced protocol is secure over offline dictionary attacks explained in the previously chapter.
2. **Resistance to MITM attack.** An enemy A may catches / sniffs on the dialog between S and U . Subsequently take hold of the message (β, C_u, e_u) transferred by U , A might impersonate and answer the message to S . Although A has the reply message e_s from S , A cannot calculate any values in e_s without having knowledge of R which is not exposed ever on the interaction. Additionally, A cannot forge a dialog to impersonate S or U without having knowledge of R . Utilizing the symmetric key R , our proposal protocol protect from the man-in-the-middle hacking explained in previous part of chapter. Furthermore, R doesn't require to be interchanged during dialog; S and U are able to get R by it calculating on each part. Hence, the proposal protocol can resist the man-in-the-middle hacking.
3. **Resistance to lost smart card attack.** Imagine A has lost U 's smart card and stored the transferred messages $(\beta, C_u, e_u, e_s, M)$ in the period of U 's past sessions. Although, thus A doesn't have knowledge of ID_u and PW_u , A cannot forge information between S and U that sends login inspection or forge SK_s and SK_u without having knowledge of PW_u , n_u and n_s . Thus, the proposal protocol can resist the lost smart card attack.

4. **Resistance to inside attack.** Hence U registers to S by maintaining $\alpha = h(b \oplus PW_u)$ substitutes for PW_u , the inner entity S cannot directly intercept PW_u . Moreover, as b is not disclosed to S , the inner entity of S cannot intercept PW_u by executing an offline dictionary hacking on α . Consequently the proposal protocol can withstand the insider attack.

We have proposed an advanced protocol with higher resistance to the attack of man-in-the-middle, attack of off-line dictionary, lost smart card hacking and inner entity attack to keep away from these attacks.

4.1.6 Comparison with Other Works

PAKE schemes (e.g., EKE, SRP), do not need to encrypted channels to preserve the password and get the additional benefit of building a interactive authenticated key of session that can be applied to protect later session. Today's PAKE schemes don't supply confidentiality protection to the user's identification. An attack of man-in-the-middle is still available over un-reliable frailty session depending on which authentication of the mutual party in implemented. Hence, it is needed for our scheme to supply protection though the user contacts to an attacker substitutes for the intended party. The disadvantage of Juang and LCW scheme is that they don't supply user's confidentiality, an essential point in e-commerce software. Our proposal scheme supply privacy preserving to the user's identification. Although, a user is cheated into executing the scheme directly with an hacker, the user's credential is never disclosed.

Table 4.1: Calculation cost related protocols

| Computation | Juang | | LCW | | Improved | |
|----------------|----------------|-------------------|------|-----------|-----------|---------------------|
| | U | S | U | S | U | S |
| Registration | - | 1H | - | 1H+1Exp | 1H+1R | 2H+1Exp |
| Login | 1H+1Enc+2R | - | 1H+1 | - | 2H+1Enc+1 | - |
| Authentication | 1H+1Enc+1Dec+1 | 1H+1Enc+2Dec+3C+2 | 1C | 2H+1Exp+2 | 1Dec+1C | 3H+1Exp+1Enc+1Dec+2 |

Communication means the number of messages transferred between the server and the user.

H: the calculation-time for a hash-function.

Exp: the calculation-time for modular-exponentiation.

Enc/Dec: the calculation-time for encryption or decryption.

C: the calculation-time for comparison.

R: the calculation-time for random-number generation.

Table 4.2: Comparison-functionality between related protocols

| Functionality | Juang | LCW | Improved |
|--------------------------------|---------|---------|----------|
| Mutual authentication | Provide | - | Provide |
| Freely changed password | - | Provide | Provide |
| Session key agreement | Provide | Provide | Provide |
| User anonymity | - | - | Provide |

We compare the cost of calculation and functionality of our proposal scheme with previously described two schemes: the scheme presented by Juang [37] and the scheme proposed by Liao [38]. As for calculation costs, the table shows the amount of calculation time performed per each server and user. These two schemes are known as effective protocols among key exchange protocols released up to date. As mentioned before, the protocol of Juang, in its elemental form, is primary protocol of Chien et al. [42] scheme. To compare with Chien et al.'s protocol, Juang's generating a key of session shared by the user and the server. The protocol by Liao' scheme. improves on the protocol of Juang in terms of efficiency. In our protocol, the focus is in particular on security against malicious user and server at a very small efficiency cost. Comparison of computation efficiency and functionality between our protocol and previous solutions indicate the functionality superiority of the proposed protocol.

4.2 An Improved Authentication and Key Exchange Scheme

The entity authentication is a really significant security system in a remote login system. There are a lot of ways for authenticating users. And one of the most useful authentication systems is the password-based authentication scheme. The user authentication protocol using password [68] is firstly proposed by Lamport in 1981. Though a lot of protocols complementing Lamport's scheme have been suggested [69-80] since then, for the common wireless LAN services all of those protocols were unsuitable.

Contrasted with normal services, the common wireless LAN services should fulfill its special features like security, roaming and billing. Particularly, security in common wireless LAN service is an important topic. There are some security conditions of the common wireless LAN services. First of all, it should guarantee anonymity of the user. If the condition is not guaranteed, a hacker can completely access the information about the respective user's location. we are called anonymous commu.system to prevent user's identity from being exposed only during communication [81]. Second, it should be possible for a server and an user to authenticate mutually.

If it is not ensured, an hacker may act as an authenticated server or user. Third, it is essential for a public unsecure wireless LAN system to fulfill the forward true confidentiality. An attacker can calculate a session private key information if it does not fulfill the forward confidentiality. In 2003, an AKE protocol complying with the above conditions [82] is proposed by Park et al.

However, in 2008, Juang et al. proved that Park et al.'s system could not satisfy the requirement that the user anonymity should be guaranteed and suggested a new protocol satisfying the condition [83]. The protocol Juang et al. suggested have an advantage to need low level calculation overhead than the previous scheme while guaranteeing the anonymity, but we find out that their proposal does not guarantee the user anonymity and is vulnerable to the stolen-verifier attack.

Furthermore, a weakness that the server has high calculation overhead will be come out. In this dissertation, we analyze the vulnerability of their scheme and suggest an enhanced protocol to guarantee the user anonymity as well as the anonymous communication. The formation of this dissertation is organized as follows. We prove out suggested protocol in Section 4.2. we talk about the security and efficiency of our protocol. Finally we will make conclusions.

4.2.1 Proposed protocol

In this chapter, we suggest an enhanced anonymous authentication and key exchange protocol. While maintaining the advantages of the existing protocol, our protocol provides the user anonymity, has low calculation overhead than Juang et al.'s protocol and is secure against the stolen-verifier attack.

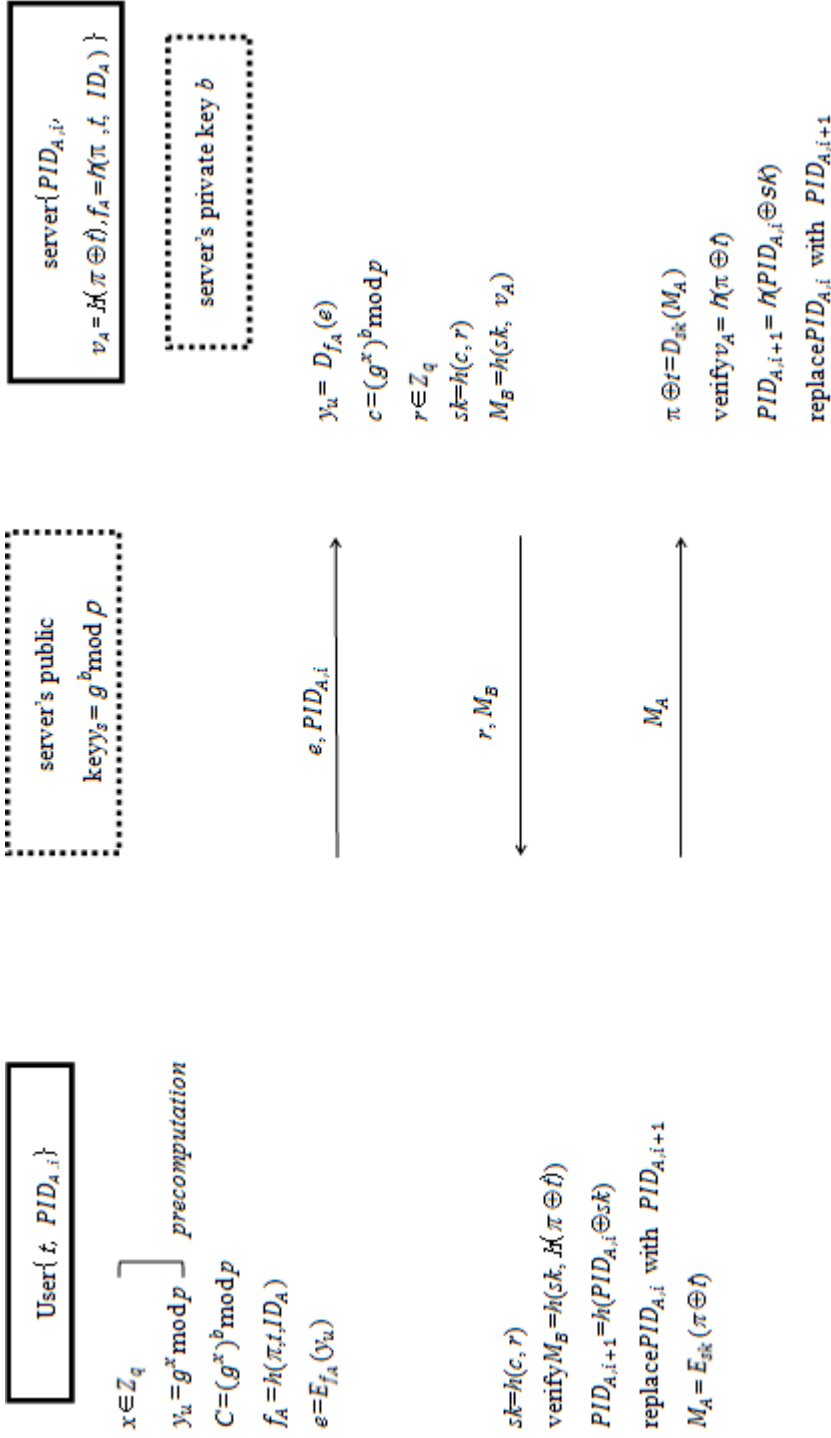


Fig. 4.6: Proposed protocol

<Notations>

- p : a large prime number
- q : a prime divisor of $(p-1)$
- g : an element of order q in Z_p^*
- b : a private key of the server
- y_s : a public key of the server ($= g^b \bmod p$)
- A : the user
- B : the server
- π : the password of the user
- t : the shared symmetric key between the user and the server for symmetric key encryption
- i : the index of the session
- ID_A : the user's identification
- $h()$: a secure one-way hash function
- $E_k()$: a symmetric encryption function with the symmetric key K
- $D_k()$: a symmetric decryption function with the symmetric key K
- sk : a session key generated by the user and the server
- $PID_{A,i}$: a temporary ID of the user ($= PID_{A,i-1} \oplus sk$)

4.2.2 Proposed protocol

<Registration stage>

A and B shares a password π and a symmetric key t . A remembers π and stores t and $PID_{A,i}$ in a smart card. B stores $PID_{A,i}$, $v = h(\pi \oplus t)$, $f = h(\pi, t, ID_A)$ in a storage. Moreover, B selects a random number b as a private key and computes $y_b = g^b \text{ mod } p$ and sets it as a public key. b must be securely stored in B 's database.

<Pre-computation stage>

A selects a random value x in Z_q and computes $y_u = g^x \text{ mod } p$. Then, to reduce the computational overhead in the authentication and key exchange stage, A calculates $c = g^x \text{ mod } p$ and stores it.

<Authentication and key exchange stage>

In this stage, a mutual authentication between a user and the server is performed and a session key is established.

(1) A computes $f_A = h(\pi, t, ID_A)$, $e = E_{f_A}(y_u)$ and sends e and $PID_{A,i}$ to B .

(2) B acquires v and f by verifying $PID_{A,i}$. Based on f , B computes $y_u = D_{f_A}(e)$. After decryption, B computes $c = (g^x)^b \text{ mod } p$ and selects a random number r . B computes $sk = h(c, r)$ and $M_B = h(sk, v)$. Now, B sends r and M_B to A .

(3) A computes a session key $sk = h(c, r)$. Now, A and B share the session key sk . Then A calculates $M_B = h(sk, h(\pi \oplus t))$ and compares it with the M_B sent by B . If so, A authenticates B as a legitimate server. To ensure the user anonymity, A computes $PID_{A,i+1} = h(PID_{A,i} \oplus sk)$ and replaces $PID_{A,i}$ stored in the smart card with $PID_{A,i+1}$. Finally A computes $M_A = E_{sk}(\pi \oplus t)$ and sends it to B .

(4) B decrypts M_A using sk and gets $\pi \oplus t$. With this value, B computes $v_A = h(\pi \oplus t)$ and verifies whether right v_A comes out, then B authenticates A as a legitimate user. Then B computes $PID_{A,i+1} = h(PID_{A,i} \oplus sk)$ and replaces $PID_{A,i}$ stored in the server's database with $PID_{A,i+1}$.

4.2.3 Efficiency and security analysis

< Efficiency analysis >

To analyze the efficiency, we assumed the following environment.

- Hash function: SHA-1 (Hash size: 160 bits)
- Symmetric key algorithm: AES (Key length: 128 bits)
- $r(\text{nonce})$: 128 bits
- $b(\text{long-term secret key})$: 128 bits
- ID length: 32 bits
- Number of users registered in the server: n

In the above environment, the following is the result of the efficiency analysis of each scheme.

Table 4.4: Efficiency analysis proposed protocol

| Schemes | User's computation cost | Server's Computation cost | Communication Cost |
|-------------------|---|---|--------------------|
| Park et al. [82] | 5 Hash+ 1 Encryption+ 2 Exponentiation | O(n) Hash+ 1 Decryption+ 1 Exponentiation | 1664 bits |
| Juang et al. [83] | 5 Hash+ 1 Encryption + 2 Exponentiation | O(n) Hash+ 1 Decryption+ 1 Exponentiation | 1684 bits |
| Proposed protocol | 4 Hash+ 2 Encryption + 2 Exponentiation | 4 Hash+ 2 Decryption+ 1 Exponentiation | 1632 bits |

In case of user's computation cost, the proposed protocol showed the least number of hash functions with 4 compared to other protocols but it has one more encryption processes added. All protocols show same number of exponentiation computation. In case of server's computation cost, especially number of hash function computation, the protocol proposed by Juang et al. must try until right *SID* comes out by substituting every π and t that the server has to $h(\pi, t, i)$ to verify the *SID*. In the worst case, for a user to verify itself in a single session, it must perform as many hash functions as the number of users registered in the server. This results in too high computational overhead. On the other hand, the proposed protocol does not have a step to substitute a specific value into a hash function and it only has to perform 4 hash function computations.

However, it performs one more decryption process than other protocols. In case of communication cost, the proposed protocol has the smallest with 1632 bits. As a conclusion, although all protocols have similar numbers of encryption/decryption, exponentiation computations, and communication cost, the proposed protocol is more efficient than the others since the computational cost of the server's hash function is very low.

< Security analysis >

The followings are the security analysis results of each scheme.

Table 4.5: Security analysis of proposed protocol

| | Park et al. [14] | Juang et al. [15] | Proposed protocol |
|---|------------------------|-------------------------|----------------------|
| Providing the user anonymity | × | × | ○ |
| Providing the mutual authentication | ○ | ○ | ○ |
| Providing the half-forward secrecy | ○ | ○ | ○ |
| Secure against the user impersonation attack | ○ | ○ | ○ |
| Secure against the server impersonation attack | ○ | ○ | ○ |
| Secure against the replay attack | ○ | ○ | ○ |
| Secure against the stolen-verifier attack | × | × | ○ |
| Secure against the guessing attack | × | ○ | ○ |

(1) **User Anonymity**: In this protocol, users' ID is not exposed during communication and it is not stored in the server but stored in temporary ID , PID forms. Therefore, it satisfies the user anonymity.

(2) **Mutual Authentication**: Since an attacker cannot know sk and $\pi \oplus t$, it cannot compute M_A and M_B . That is, only legitimate users and the server can compute M_A and M_B . Therefore, users and the server can mutually authenticate each other through M_A and M_B .

(3) **Half-forward Secrecy**: Half Forward Secrecy means the loss of one side's long-lived key should not be damaging to the previous sessions[16]. Let's assume that an attacker figured out a user's secret information (ID, π, t) . The attacker may find out $g^x \bmod p$ by decrypting $E_A(g^x \bmod p)$. However, since the attacker does not know b , it cannot compute c , and cannot compute the session key. Therefore, the user side satisfies the forward secrecy. Again, let's assume the attacker find out the server's secret information (ID, π, t, b) . The attacker may find out $g^x \bmod p$ by decrypting $E_T(g^x \bmod p)$, and since it knows b , it can compute c . Finally, the attacker can compute the session key. Therefore, since this protocol satisfies the forward secrecy only from the user side, it satisfies the half-forward secrecy.

(4) **User Impersonation Attack**: For an attacker impersonates as the legitimate user, it must transmit correct M_A to the server. However, since the attacker cannot compute the session key sk without knowing c , it cannot figure out the correct M_B . Therefore, this protocol is secure against the user impersonation attack.

(5) **Server Impersonation Attack:** For an attacker impersonates as the legitimate server, it must send correct M_B . However, since the attacker cannot compute the session key sk without knowing c , it cannot figure out the correct M_B . Therefore, this protocol is secure against the server impersonation attack.

(6) **Replay Attack:** A user randomly creates x on each session. Therefore, a different value is created every time for the e . Moreover, since a different value is created each time for M_A and M_B due to a random value r , this protocol is secure against the replay attack.

(7) **Stolen-verifier Attack:** An attacker is able to get PID , $v = h(\pi \oplus t)$ and $f = h(\pi, t, ID_A)$ by attacking the server's database. However, it cannot find out users' ID , password and symmetric key only with these information. An attacker may impersonate as the legitimate user based on the acquired information using the stolen-verifier attack. In this case, an attacker should find out M_A , but since it cannot compute $(\pi \oplus t)$, it is unable to compute M_A . Therefore, an attacker cannot impersonate as the legitimate user. An attacker may also impersonate as a legitimate server based on the acquired information through the stolen-verifier attack, but it must know the secret key b to succeed an attack. However, b is secret information, which cannot be acquired through the stolen verifier attack, so an attacker cannot impersonate as the legitimate server. Therefore, this protocol is secure against the stolen-verifier attack.

(8) **Guessing Attack:** The secret information for an attacker to try guessing attack is v , M_A, M_B and f . To analogize the password π based on those information, it must know t , ID and c . However, there is no way for an attacker to find out those values. Moreover, the attacker may perform the guessing attack by trying XOR with the acquired values. If there is secret information separately with low entropy such as password or ID in the combined values by the attacker, the attacker may find out the password by performing the guessing attack. However, as a result of analyzing the combinations of all information that an attacker can get, we confirmed that there is no information combination separately including password or ID . We also confirmed that some other information don't include secret information at all and they are nonce or meaningless information. Therefore, this protocol is secure against the guessing attack. The table 4.4 and 4.5 in the Appendix describe possible combinations of the values to break our protocol using a guessing attack.

(9) **Resistance to MITD attack.** A hacker A may intercept on the communication line between sessions (*user* and *server*). After this acting the message $(e, PID_{a,i})$ sent by *user*, A may impersonate and replay the $(e, PID_{a,i})$ to S . Even if A has the response message (r, M_B) from *server*, A can't extract any values in M_B without knowing t which A can't know the value t . In addition, A can't forge a message to impersonate *user* or *server* without knowing $v_A = h(\pi \oplus t)$. Using the symmetric key t , our proposed scheme prevents the man-in-the-middle attack described in previously section. *User* and *server* can get t by it computing on each side. Thus, the proposed scheme can withstand the man-in-the-middle attack.

4.2.4 Summery

In our thesis, we have proved that the several schemes were vulnerable to the stolen-verifier attack, MITD attack and other attacks did not satisfy the user anonymity, and proposed an improved protocol. The proposed protocol ensures the user anonymity, is secure against the stolen-verifier attack and satisfies the half-forward secrecy. In the future studies on improved protocols on efficiency based on the proposed protocol should be conducted.

5. Conclusion

This Thesis investigated the security of related authenticated key exchange protocols for insecure network environments. We provide AKE security fulfils for analyzing of the AKE protocols. We reviewed and found out that published AKE protocols suffered from serious weaknesses, so that we propose improvements of the AKE protocols. In this part, we summarize the contribution of this work and provide our conclusions.

Widely used of remote wireless access and computing, roaming and user authentication has emerged as a significant issue for wireless communication or network operators and end to end users. The design difficulty for AKE protocol is the multi-party users, scalability, limited battery supply to the wireless devices, and many potential adversaries. This thesis begins with the security analysis of previous solutions for AKE to bring out the security requirements inherent in designing AKE protocols. AKE protocol's security is not static because new attack methods and computing powers more and more rapidly develops and a previously secure system to fail. So that system designers have to think about both their system design and potential adversary attacks. We design the following issues:

Firstly, we provide authenticated key exchange security targets and property requirements for analyzing the security of authenticated key exchange protocols used to compare previously related works and our protocols.

Secondly, we evaluate the security of previous related works for AKE facing

security threats on the authenticated key exchange (AKE) for session initiation protocol, hash-based protocol and one-time password authenticated key exchange (AKE) protocol. We pointed out that general potentially attacks, drawbacks of the protocols and propose improvements using encryption, hash chain and one-time-password so that derives lower computation and better secure models.

Third, we analyze AKA protocols and propose improved authenticated key agreement (AKA) protocols with efficiency and secure features over public network environments. Our schemes have properties which are based on symmetric encryption, cryptographic hash function such as firstly, exquisite AKA protocol preserving user privacy, secondly, efficient and secure AKA protocol preserving user privacy.

From these observations, we provide fundamental AKE security goals and we review and analyze the previously recent protocols for AKE facing security threats. Lastly, we design efficient and secure AKA protocols for general purpose over public network environments. Also we analyze the security and efficiency for our proposal.

Bibliography

References

- [1] Nam J, Paik J, Kang H, Kim U, Won D, “An off-line dictionary attack on a simple three-party key exchange protocol”, IEEE Communication Letter, pp. 205-207, 2009.
- [2] Jeong H, Won D, Kim S, “Weakness and improvement of secure hash-based strong password authentication protocol”, Information Science, Eng 26(5): 1845-1858, 2010.
- [3] Zhu J, Ma J, “A new authentication scheme with anonymity for wireless environments”, IEEE Trans Consumer Electron 50(1) 13(3):230-234, 2009.
- [4] Lee CC, Hwang MS. Liao, “IE Security enhancement on new authentication scheme with anonymity for wireless environments”, IEEE Transaction and Electronic, 53(5):1683-1687, 2006.
- [5] Wu CC, Lee WB, Tsaur WJ, “A secure authentication scheme with anonymity for wireless environments”, IEEE Comu. Letter 12(10): 72-723, 2008.
- [6] Cui X, Qin X (2011), “An enhanced user authentication scheme for wireless communications”, IEICE Trans Info Syst E94-D(1): 155-157, 2011.
- [7] H. F. Huang and W. C. Wei, “A New Efficient Authentication Scheme for Session Initiation Protocol”, Joint Conference Info. Sciences, October 2006.

- [8] M Handley, A. Schulzrinne, E. Schooler, and J. Rosenberg, “SIP: The Session Initiation Protocol”, IETF RFC 2543, March 1999.
<http://www.ietf.org/rfc/rfc2543.txt>
- [9] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, “The Session Initiation Protocol”, <http://www.ietf.org/rfc/rfc3261.txt>
- [10] William Stallings, “Network Security Essentials, Applications and Standards”, Prentice-Hall, 2000.
- [11] C. C. Yang , R. C. Wang, W. T. Lin, “Secure Authentication scheme for Session Initiation Protocol”, Computers & Security, vol.24, no.5, pp 381-386, 8.2005.
- [12] M. Bellare, D. Pointcheval, P. Rogaway, “Authenticated Key Exchange Secure against Dictionary Attacks”, In Advances in Cryptology-Eurocrypt, LNCS 1807, pp. 139-155, May 2000.
- [13] J. Smith and F. Weingarten, “Research challenges for the next generation internet”, Report from the Workshop on Research Directions for NGI, May 2007.
- [14] C. Mitchell, “Security for Mobility”, IEEE press, 2004.
- [15] J. Kohl and C. Neuman, “The Kerberos network authentication service (v5)”, Internet Request for Comments 1510, 1993.
- [16] S. Bellovin and M. Merritt, “Limitations of the Kerberos authentication system, ACM Computer Communications Review”, vol. 20, no. 5, pp. 119-132, 1990.
- [17] H. Y. Chien and J. K. Jan, “Robust and simple authentication protocol”, The Journal of Computer, Feb 2003.
- [18] A. Fox and S. Gribble, “Security on the movie, indirect authentication using Kerberos”, In Proceedings of the 2nd ACM International

Conference on Mobile Computing and Networking, pp. 154-164, 1996.

- [19] R. Ganesan: Yaksha, “Augmenting Kerberos with public key cryptography”, In Proceedings of Symposium on Network and Distributed System Security, IEEE Computer Society, pp. 132 - 143, 1995.
- [20] M. Sirbu and J. Chuang, “Distributed authentication in Kerberos using public key cryptography”, In Proceedings of Symposium on Network and Distributed System Security, IEEE, pp.134-141, 1997.
- [21] M. Bellare and P. Rogaway, “Entity Authentication and key distribution”, In advances in Cryptology, CRYPTO’93, LNCS 773, pp. 232-249, 1994.
- [22] K. Nyberg and R. Rueppel, “Weaknesses in some recent key agreement protocols”, Electronics letters, vol. 30, pp. 26-27, 1994.
- [23] Y. Yacobi, “A key distribution paradox, In Proceedings of the 10th International Cryptology Conference on Advances in Cryptology, vol. 537, pp. 268-73, 1991.
- [24] I. Kao and R. Chow, “An efficient and secure authentication protocol using uncertified keys”, ACM Operating Systems Review, vol. 29, pp. 14-21.1995
- [25] S. Shieh, F. Ho, and Y. Huang, “An efficient authentication protocol for mobile networks”, The Journal of Information Science and Engineering, vol. 15, pp. 505-520, 1999.
- [26] Q. Tang and C. Mitchell, “Cryptanalysis of a hybrid authentication protocol for large mobile networks”, The Journal of Systems and Software, vol. 79, pp. 496-501, 2006

- [27] W. Shi, I. Jang and H. Yoo, "A provable secure authentication protocol given forward secure session key", In Proceedings of the 10th Asia Pacific Web Conference, vol.4976, pp. 309-318, 2008
- [28] M. S. Hwang and L. H. Li, "A new remote user authentication scheme using smart cards", IEEE transactions on Consumer Electronics, vol. 28, pp. 241-252, 2005
- [29] S. Bellare and A. Merritt, "Encrypted Key Exchange Passwordbased Protocols Secure against dictionary attacks", In Proceedings of the IEEE Symposium on Security and Privacy, pp. 7-84, May 1992.
- [30] V. Boyko, P. MacKenzie, S. Patel, "Provably secure password authenticated key exchange using Diffie-Hellman", In Advances in Cryptology Eurocrypt, LNCS 1807, pp.156-171, May 2000.
- [31] D. P. Jablon, "Strong password only authenticated key exchange", ACM SIGCOMM Computer Communication Review, vol. 26. No. 5, pp.5-26, 1996
- [32] T. Wu, "The Secure Remote Password protocol, Internet Society", Network and Distributed System Security Symposium, pp. 97-111, 1998
- [33] C. C. Yang, R. C. Wang, W. T. Lin, "Secure Authentication Scheme for Session Initiation Protocol", Computers & Security, vol. 24, no. 5, pp.381-386, August 2005
- [34] G. Yang, D. S. Wong, and X. Deng, "Two-factor mutual Authentication based on smart cards and passwords", The Journal of Computer and System Sciences, vol. 74. No. 7, pp. 1160-1172, 2008

- [35] L. Lamport, "Password authentication with insecure Communication", *Communications of the ACM*, vol. 24, no. 11, pp. 770-772, Nov. 1981.
- [36] M. S. Hwang, C. C. Lee, Y. L. Tang, "A simple remote user authentication scheme", *Mathematical and Computer Modeling*, vol. 36, pp. 103-107, 2002.
- [37] W. S. Juang; "Efficient password authentication key agreement using smart cards", *Computers and Security*, vol.23, no. 2. Pp.167-173, 2004.
- [38] C. H. Liao, H. C. Chen, and C. T. Wang, "An exquisite mutual authenticated scheme with key agreement using smart card", *The International Journal of Computing and Informatics*, vol. 33. No. 2, pp.125-132, 2009.
- [39] M. Kim, S. Kim, and D. Won, "An Exquisite Authenticated scheme with key agreement Preserving User Anonymity", *International Conference on Web Information Systems and Mining(WISM 2010)*, LNCS 6318, pp. 244-253, Oct. 2010.
- [40] P. Kocher, J. Jaffe, and B. June: Differential power analysis, In *Advances in Cryptology- CRYPTO '99*, pp. 388-397, 1999.
- [41] T. S. Messeres, E. A. Dabbish, and R. H. Sloan, "Examining smart card security under the threat of power analysis attacks", *IEEE Transactions on Computer*, vol. 51, no. 5, pp.541-552, 2002.
- [42] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "An efficient & practical solution to remote authentication Smart card", *Computers and Security*, vol. 21, no. 4, pp.372-375, 2002
- [43] S. T. Wu and B.C. Chieu, "A note on a user friendly remote user authentication scheme with smart cards", *IEICE Transactions Fundamentals*, vol. 87-A, no. 8, pp.2180-2181, 2004

- [44] M. K. Khan, "An efficient and secure remote mutual authentication scheme with smart cards", Information Symposium on Biometrics and Security Technologies, pp. 1-6, April. 2008.
- [45] A. J. Menezes, P. C. Oorschot, and S. A. Vanston, "Handbook of applied cryptography", CRC Press, 1997
- [46] C. Boyd and A. Mathuria, "Protocols for Authentication and Key establishment", Springer-Verlag, 2003.
- [47] K. R. Choo, "Secure Key Establishment, In Advances in Information Security", Springer-Verlag, 2009.
- [48] W. Diffie, M. Wiener, P. Van Oorschot, "Authentication and Authenticated Key Exchanges, Designs, Codes, and Cryptography", kluwer Academic publishers, pp. 107-125, 1992.
- [49] W. Mao, "Modern cryptography; Theory and Practice", Prentice Hall PTR, 2003.
- [50] L. Lamport, "Password authentication with insecure communication", Comm. of the ACM, vol.24, no.11, pp. 770-772, 1981.
- [51] R. E. Lennon, S. M. Matyas, C. H. Mayer, "Cryptographic authentication of time-invariant quantities", IEEE Trans, Commu. COM-29, vol.6, pp. 773-777, 1981.
- [52] S. M. Yen, K. H. Liao, "Shared authentication token secure against replay and weak key attack", Information Proceeding Letters, pp.78-80, 1997
- [53] C. C. Chang, T. C. Wu, "Remote password authentication with smart cards", IEEE proceedings, Part E. Computer and Digital Techniques, vol. 138, no.3, pp.165-168, 1991
- [54] S. J. Wang, J. F. Chang, "Smart card based secure password authentication scheme", Computers and Security, vol.15, no.3, pp. 231-237, 1996.

- [55] W. H. Yang, S. P. Shieh, "Password authentication schemes with smart card", *Computer and Security*, vol.18, no.8, pp. 727-733, 1999.
- [56] M. S. Hwang, L. H. LI, "A new remote user authentication scheme using smart card", *IEEE Transaction on Consumer Electronics*, vol. 46, no.1, pp.8-30, 2000.
- [57] H. M. Sun, "An efficient remote user authentication scheme using smart cards", *IEEE Transaction on Consumer Electronics*, vol. 46, no.4, pp.958-961, 2000
- [58] H. Y. Chien, J. K. Jan, Y. M. Tseng, "An efficient and practical solution to remote authentication: smart card", *Computer and Security*, vol. 21. Pp.373-375, 2002.
- [59] W. C. KU, S. M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards", *IEEE Transaction on Consumer Electronics*, vol. 50, no.2, pp.204-207, 2004.
- [60] X.M. wang, W. F. Zhang, J. S. Zhang, M. K. Khan, "Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards", *Computer Standards and Interfaces*, vol.29. no.5, pp.507-512, 2007.
- [61] E. J. Yoon, E. K. Ryu, K. Y. Yoo, "Further improvement of an efficient password based remote user authentication scheme using smart cards", *IEEE Transaction on Consumer Electronics*, vol. 50, no.2, pp.612-614, 2007.
- [62] Nam J, Paik J, Kang H, Kim U, Won D, "An off-line dictionary attack on a simple three-party key exchange protocol", *IEEE Commun. Letters* pp 205-207. 2009.

- [63] Jeong H, Won D, Kin S, “Weakness and improvement of secure hash-based strong password authentication protocol”, J Inform Sci. Eng, 26(5): 1845-1858. 2010.
- [64] Zhu J, Ma J, “A new authentication scheme with anonymity for wireless environments”, IEEE Trans Consum Electron 50(1) 13(3):230-234. 2009.
- [65] Lee CC, Hwang MS. Liao IE, “Security enhancement on a new authentication scheme with anonymity for wireless environments”, IEEE Trans Ind Electronic, 53(5):1683-1687(2006).
- [66] Wu CC, Lee WB, Tsaur WJ, “A secure authentication scheme with anonymity for wireless environments”, IEEE Communication Letters, 12(10): 772-773, 2008.
- [67] Cui X, Qin X, “An enhanced user authentication scheme for wireless communications”, IEICE Trans Info Syst E94-D(1): 155-157, 2011.
- [68] Lamport L, “Password authentication with insecure communication,” Communications of the ACM, 1981;24(11):770–772.
- [69] Awasthi A, Lal S, “A remote user authentication scheme using smart cards with forward secrecy,” IEEE Transaction Consumer Electronic,;49(4):1246–1248, 2003.
- [70] Awasthi A, Lal S. “An enhanced remote user authentication scheme using smart cards,” IEEE Trans. Consumer Electronic, 2004;50(2):583–586.
- [71] Juang W, “Efficient password authenticated key agreement using smart card,” Computers & Security, 2004; 23:167–73.
- [72] Ku W, Chen S, “Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards,” IEEE Trans. Consumer Electronic, 2004;50(1):204–7.

- [73] Kumar M, "New remote user authentication scheme using smart cards," IEEE Trans. Consumer Electronic, 2004; 50(2):597–.600.
- [74] Kwon T, Park Y, Lee H, "Security analysis and improvement of the efficient password-based authentication protocol," IEEE Communications Letters, 2005; 9(1):93–.5.
- [75] Park Y, Park S, "Two factor authenticated key exchange (TAKE) protocol in public wireless LANs," IEICE Trans. Communications 2004; E87-B(5):1382–.5.
- [76] Sun H, "An efficient use authentication scheme using smart cards," IEEE Trans. Consumer Electronic, 2000; 46(4):958–.61.
- [77] Wang X, Zhang W, Zhang J, Khan M. "Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards," Computer Standards & Interfaces, 2007; 29(5):507–.12.
- [78] Yang C, Hwang M, "Cryptanalysis of simple authenticated key agreement protocols," IEICE Trans. Communications, 2004; E87-A(8):2174–.6.
- [79] Yang C, Wang R, "Cryptanalysis of a user friendly remote authentication scheme with smart cards," Computer Security, 2004; 23:425–.7.
- [80] Zhenchuan Chai, Zhenfu Cao, Rongxing Lu, "Efficient Password-Based Authentication and Key Exchange Scheme Preserving User Privacy," Wireless Algorithms, Systems, and Applications 2006, Vol. 4138, pp. 467-477.
- [81] Young Man PARK, Sang Kyu PARK, "Two factor authenticated key exchange(TAKE) protocol in public wireless LANs," IEICE Trans. Communications, 2004, E87-B(5), pp. 1382-1385.

- [82] Wen-Shenq Juang, Jing-Lin Wu, "Two efficient two-factor authenticated key exchange protocols in public wireless LANs," Computers and Electrical Engineering, 2008, Vol.10, pp. 1-8.
- [83] Tzu-Chang YEH, Hsiao-Yun SHEN, Jing-Jang HWANG, "A Secure One-Time Password Authentication Scheme Using Smart Cards," 2002, Vol.E85-B No.11, pp.2515-2518.

초 록

인증키 교환 프로토콜은 통신상의 안전하지 못한 부분을 해결 하기 위해 설계되며, 해커 등 일명 악의적 마음을 품은 적들은 공중 네트워크 상의 안전하지 않은 취약점을 이용하여 공격한다.

불안전 문제들을 개선해 왔음에도 불구하고 정보보안은 주요 연구분야로 남게 된다. 더욱이, 오늘날 사용자 인증기법의 익명성은 매우 중요하다. 본 논문은 다양한 인증기법들을 논하고 이들 논문들의 취약점을 논하고 논문에 수록되지는 않았지만 선형공격, 역공학 등의 기법을 동원 연구하여 개선된 인증과 키 교환 기법을 제안한다.

주요어: (암호학, 선형공격, 상호인증, 취약점, 역공학익명, 키교환)

학번:(2004-30103)